

```

library(flexsurv)

## Loading required package: survival
## Loading required package: splines

library(boot)

##
## Attaching package: 'boot'
##
## The following object is masked from 'package:survival':
##
## aml

library(randomForestSRC)

## Loading required package: parallel
##
## randomForestSRC 1.5.5
##
## Type rfsrc.news() to see new features, changes, and bug fixes.
##

library(timeROC)

## Loading required package: pec
## Loading required package: mtnorm
## Loading required package: timereg

library(risksetROC)

## Loading required package: MASS

library(ggplot2)

## Loading required package: methods

library(RColorBrewer)

```

```

pal = brewer.pal(4, "Dark2")
names(pal) = c("gg", "km0", "mskcc.post", "mskcc.pre")

```

## 1 Preparation

Construct a *\*preoperative\** function based on the Brennan nomogram. The preoperative nature will mean that most prognostic components will need to be marginalized out.

So the preoperative MSKCC score would be:

$$S = 1.4 + 6.1 + 0.8 + 18.2 + 18.9 + 15 + 9 + 15 * Back.pain + 3 * Weight.Loss + -2/15 * Age + 12 + 3 [Sex = M] + 51 [Hea$$

(1)

```

fit.mskcc = list(
  inputs = list(
    History.Diagnosis.AgeAt = list(
      margins = data.frame(value = 65, fraction = 1),

```

Variable	Preoperative?	Available?	Marginals
Age	Yes	Yes	Linear. 90 =>0, 30 =>8. Therefore $f(x) = -2/15(x - 90) = -2/15x + 12$
Sex	Yes	Yes	Male risk delta 3
Portal Vein	NO		14.4% YES, risk delta 10, marginal 1.4
Splenectomy	NO		9.9% YES, risk delta 62, marginal 6.1
Margin of resection	NO		20.7% POS, risk delta 4, marginal 0.8
Head.vs.Other	Yes	Yes	Head risk delta 51
Differentiation	NO		14.2% Well, risk delta 0, marginal 0
			56.4% Mod, risk delta 14, marginal 7.9
			29.5% Poor, risk delta 35, marginal 10.3. Overall marginal 18.2
Posterior.margin	NO		86.0% POS, risk delta 22, marginal 18.9
Numb.pos.nodes	NO		Mean 2.1, approx marginal 15
Numb.neg.nodes	NO		Mean 16.9, approx marginal 9
Back.pain	Yes	NO	13.7% YES, risk delta 15, marginal 2.0
T.stage	Yes	Yes	
Weight Loss	Yes	NO	53.7% YES, risk delta 3, marginal 1.6
Max.path.axis	Yes	Yes	

```

scorefunc = function(x) { x = x; -2/15*pmin(pmax(x, 0), 90) + 12 },
Patient.Sex = list(
  margins = data.frame(value = c("M", "F"), fraction = c(0.501, 1-0.501)),
  scorefunc = function(x) { 3*I(x == "M") }),
Portal.Vein = list(
  margins = data.frame(value = c(TRUE, FALSE), fraction = c(0.144, 1-0.144)),
  scorefunc = function(x) { 10*I(x == TRUE) }),
Splenectomy = list(
  margins = data.frame(value = c(TRUE, FALSE), fraction = c(0.099, 1-0.099)),
  scorefunc = function(x) { 62*I(x == TRUE) }),
Treat.MarginPositive = list(
  margins = data.frame(value = c(TRUE, FALSE), fraction = c(0.207, 1-0.207)),
  scorefunc = function(x) { 4*I(x == TRUE) }),
Path.LocationBody = list(
  margins = data.frame(value = c(FALSE, TRUE), fraction = c(0.894, 1-0.894)),
  scorefunc = function(x) { 51*I(x == TRUE) }),
Path.Differentiation = list(
  margins = data.frame(value = c("1", "2", "3", "4"), fraction = c(0.142, 0.564, 1-0.142-0.564)),
  scorefunc = function(x) { 14*I(x == "2") + 35*I(x == "3") + 35*I(x == "4") }),
Posterior.Margin = list(
  margins = data.frame(value = c(TRUE, FALSE), fraction = c(0.86, 1-0.86)),
  scorefunc = function(x) { 22*I(x == TRUE) }),
Path.LN.Involved = list(
  margins = data.frame(value = 2.1, fraction = 1),
  scorefunc = function(x) {
    x = pmin(40, pmax(x, 0))
    fitfun = splinefun(c(0, 1, 2, 3, 4, 10, 15, 20, 25, 30, 35, 40), c(0, 14.56, 24.56, 24.56, 24.56, 24.56, 24.56, 24.56, 24.56, 24.56, 24.56, 24.56, 24.56))
    fitfun(x)
  }),
Path.LN.Negative = list(
  margins = data.frame(value = 16.9, fraction = 1),
  scorefunc = function(x) { (pmin(pmax(x, 0), 90)-90)*-11/90 }),
Back.pain = list(
  margins = data.frame(value = c(TRUE, FALSE), fraction = c(0.137, 1-0.137)),
  scorefunc = function(x) { 15*I(x == TRUE) }),

```

```

Stage.pT.Simplified = list(
  margins = data.frame(value = c("T1", "T2", "T34"), fraction = c(0.037, 0.119, 1-0.037-0.119)),
  scorefunc = function(x) { 36*I(x == "T1") + 11*I(x == "T34") },
  # The following matches the original Brennan nomogram, but was not used as there are too
  # tumours in either the NSWPCN *or* the MSKCC cohorts -- how the T4 coefficient was even
  # I'll never know. The T34 coefficient of 11 was arrived at as (0.828*10+(1-0.037-0.119)*
  # being a frequency-weighted average of the T3 and T4 coefficients.
  # margins = data.frame(value = c("T1", "T2", "T3", "T4"), fraction = c(0.037, 0.119, 0.037, 0.037)),
  # scorefunc = function(x) { 36*I(x == "T1") + 10*I(x == "T3") + 63*I(x == "T4") },
Weight.loss = list(
  margins = data.frame(value = c(TRUE, FALSE), fraction = c(0.537, 1-0.537)),
  scorefunc = function(x) { 3*I(x == TRUE) },
Path.Size = list(
  margins = data.frame(),
  scorefunc = function(x) {
    x = pmin(16, pmax(x, 0))
    fitfun = splinefun(c(0, 1, 2, 3, 4, 6, 8, 10, 12, 14, 16), c(0, 29.74, 59.48, 89.22, 118.96, 148.69, 178.43, 208.17, 237.91, 267.65, 297.39))
    fitfun(x)
  } ),
outputs = list(
  DSS12mo = function(s) {
    x = pmax(50, pmin(350, s))
    fitfun = splinefun(c(79.0323, 115.02, 165.524, 197.278, 221.774, 242.339, 261.081, 280.735, 300.389, 320.043, 339.697, 359.351))
    y = fitfun(x)
    pmax(0, pmin(1, y))
  },
  DSS24mo = function(s) {
    x = pmax(50, pmin(350, s))
    fitfun = splinefun(c(71.1694, 97.7823, 129.536, 153.73, 174.294, 193.347, 211.791, 230.844, 249.897, 268.95, 288.003, 307.056, 326.109, 345.162, 364.215, 383.268, 402.321, 421.374, 440.427, 459.48, 478.533, 497.586, 516.639, 535.692, 554.745, 573.798, 592.851, 611.904, 630.957, 650.01, 669.063, 688.116, 707.169, 726.222, 745.275, 764.328, 783.381, 802.434, 821.487, 840.54, 859.593, 878.646, 897.699, 916.752, 935.805, 954.858, 973.911, 992.964, 1012.017, 1031.07, 1050.123, 1069.176, 1088.229, 1107.282, 1126.335, 1145.388, 1164.441, 1183.494, 1202.547, 1221.6, 1240.653, 1259.706, 1278.759, 1297.812, 1316.865, 1335.918, 1354.971, 1374.024, 1393.077, 1412.13, 1431.183, 1450.236, 1469.289, 1488.342, 1507.395, 1526.448, 1545.501, 1564.554, 1583.607, 1602.66, 1621.713, 1640.766, 1659.819, 1678.872, 1697.925, 1716.978, 1736.031, 1755.084, 1774.137, 1793.19, 1812.243, 1831.296, 1850.349, 1869.402, 1888.455, 1907.508, 1926.561, 1945.614, 1964.667, 1983.72, 2002.773, 2021.826, 2040.879, 2059.932, 2078.985, 2098.038, 2117.091, 2136.144, 2155.197, 2174.25, 2193.303, 2212.356, 2231.409, 2250.462, 2269.515, 2288.568, 2307.621, 2326.674, 2345.727, 2364.78, 2383.833, 2402.886, 2421.939, 2440.992, 2460.045, 2479.098, 2498.151, 2517.204, 2536.257, 2555.31, 2574.363, 2593.416, 2612.469, 2631.522, 2650.575, 2669.628, 2688.681, 2707.734, 2726.787, 2745.84, 2764.893, 2783.946, 2802.999, 2822.052, 2841.105, 2860.158, 2879.211, 2898.264, 2917.317, 2936.37, 2955.423, 2974.476, 2993.529, 3012.582, 3031.635, 3050.688, 3069.741, 3088.794, 3107.847, 3126.9, 3145.953, 3165.006, 3184.059, 3203.112, 3222.165, 3241.218, 3260.271, 3279.324, 3298.377, 3317.43, 3336.483, 3355.536, 3374.589, 3393.642, 3412.695, 3431.748, 3450.801, 3469.854, 3488.907, 3507.96, 3527.013, 3546.066, 3565.119, 3584.172, 3603.225, 3622.278, 3641.331, 3660.384, 3679.437, 3698.49, 3717.543, 3736.596, 3755.649, 3774.702, 3793.755, 3812.808, 3831.861, 3850.914, 3869.967, 3889.02, 3908.073, 3927.126, 3946.179, 3965.232, 3984.285, 4003.338, 4022.391, 4041.444, 4060.497, 4079.55, 4098.603, 4117.656, 4136.709, 4155.762, 4174.815, 4193.868, 4212.921, 4231.974, 4251.027, 4270.08, 4289.133, 4308.186, 4327.239, 4346.292, 4365.345, 4384.398, 4403.451, 4422.504, 4441.557, 4460.61, 4479.663, 4498.716, 4517.769, 4536.822, 4555.875, 4574.928, 4593.981, 4613.034, 4632.087, 4651.14, 4670.193, 4689.246, 4708.299, 4727.352, 4746.405, 4765.458, 4784.511, 4803.564, 4822.617, 4841.67, 4860.723, 4879.776, 4898.829, 4917.882, 4936.935, 4955.988, 4975.041, 4994.094, 5013.147, 5032.2, 5051.253, 5070.306, 5089.359, 5108.412, 5127.465, 5146.518, 5165.571, 5184.624, 5203.677, 5222.73, 5241.783, 5260.836, 5279.889, 5298.942, 5317.995, 5337.048, 5356.101, 5375.154, 5394.207, 5413.26, 5432.313, 5451.366, 5470.419, 5489.472, 5508.525, 5527.578, 5546.631, 5565.684, 5584.737, 5603.79, 5622.843, 5641.896, 5660.949, 5679.999, 5699.05, 5718.101, 5737.152, 5756.203, 5775.254, 5794.305, 5813.356, 5832.407, 5851.458, 5870.509, 5889.56, 5908.611, 5927.662, 5946.713, 5965.764, 5984.815, 6003.866, 6022.917, 6041.968, 6061.019, 6080.07, 6099.121, 6118.172, 6137.223, 6156.274, 6175.325, 6194.376, 6213.427, 6232.478, 6251.529, 6270.58, 6289.631, 6308.682, 6327.733, 6346.784, 6365.835, 6384.886, 6403.937, 6422.988, 6442.039, 6461.09, 6480.141, 6499.192, 6518.243, 6537.294, 6556.345, 6575.396, 6594.447, 6613.498, 6632.549, 6651.6, 6670.651, 6689.702, 6708.753, 6727.804, 6746.855, 6765.906, 6784.957, 6804.008, 6823.059, 6842.11, 6861.161, 6880.212, 6899.263, 6918.314, 6937.365, 6956.416, 6975.467, 6994.518, 7013.569, 7032.62, 7051.671, 7070.722, 7089.773, 7108.824, 7127.875, 7146.926, 7165.977, 7185.028, 7204.079, 7223.13, 7242.181, 7261.232, 7280.283, 7299.334, 7318.385, 7337.436, 7356.487, 7375.538, 7394.589, 7413.64, 7432.691, 7451.742, 7470.793, 7489.844, 7508.895, 7527.946, 7546.997, 7566.048, 7585.099, 7604.15, 7623.201, 7642.252, 7661.303, 7680.354, 7699.405, 7718.456, 7737.507, 7756.558, 7775.609, 7794.66, 7813.711, 7832.762, 7851.813, 7870.864, 7889.915, 7908.966, 7928.017, 7947.068, 7966.119, 7985.17, 8004.221, 8023.272, 8042.323, 8061.374, 8080.425, 8099.476, 8118.527, 8137.578, 8156.629, 8175.68, 8194.731, 8213.782, 8232.833, 8251.884, 8270.935, 8289.986, 8309.037, 8328.088, 8347.139, 8366.19, 8385.241, 8404.292, 8423.343, 8442.394, 8461.445, 8480.496, 8499.547, 8518.598, 8537.649, 8556.7, 8575.751, 8594.802, 8613.853, 8632.904, 8651.955, 8671.006, 8690.057, 8709.108, 8728.159, 8747.21, 8766.261, 8785.312, 8804.363, 8823.414, 8842.465, 8861.516, 8880.567, 8899.618, 8918.669, 8937.72, 8956.771, 8975.822, 8994.873, 9013.924, 9032.975, 9052.026, 9071.077, 9090.128, 9109.179, 9128.23, 9147.281, 9166.332, 9185.383, 9204.434, 9223.485, 9242.536, 9261.587, 9280.638, 9299.689, 9318.74, 9337.791, 9356.842, 9375.893, 9394.944, 9413.995, 9433.046, 9452.097, 9471.148, 9490.199, 9509.25, 9528.301, 9547.352, 9566.403, 9585.454, 9604.505, 9623.556, 9642.607, 9661.658, 9680.709, 9699.76, 9718.811, 9737.862, 9756.913, 9775.964, 9795.015, 9814.066, 9833.117, 9852.168, 9871.219, 9890.27, 9909.321, 9928.372, 9947.423, 9966.474, 9985.525, 10004.576, 10023.627, 10042.678, 10061.729, 10080.78, 10099.831, 10118.882, 10137.933, 10156.984, 10176.035, 10195.086, 10214.137, 10233.188, 10252.239, 10271.29, 10290.341, 10309.392, 10328.443, 10347.494, 10366.545, 10385.596, 10404.647, 10423.698, 10442.749, 10461.8, 10480.851, 10499.902, 10518.953, 10538.004, 10557.055, 10576.106, 10595.157, 10614.208, 10633.259, 10652.31, 10671.361, 10690.412, 10709.463, 10728.514, 10747.565, 10766.616, 10785.667, 10804.718, 10823.769, 10842.82, 10861.871, 10880.922, 10899.973, 10919.024, 10938.075, 10957.126, 10976.177, 10995.228, 11014.279, 11033.33, 11052.381, 11071.432, 11090.483, 11109.534, 11128.585, 11147.636, 11166.687, 11185.738, 11204.789, 11223.84, 11242.891, 11261.942, 11280.993, 11299.994, 11319.045, 11338.096, 11357.147, 11376.198, 11395.249, 11414.3, 11433.351, 11452.402, 11471.453, 11490.504, 11509.555, 11528.606, 11547.657, 11566.708, 11585.759, 11604.81, 11623.861, 11642.912, 11661.963, 11681.014, 11700.065, 11719.116, 11738.167, 11757.218, 11776.269, 11795.32, 11814.371, 11833.422, 11852.473, 11871.524, 11890.575, 11909.626, 11928.677, 11947.728, 11966.779, 11985.83, 12004.881, 12023.932, 12042.983, 12062.034, 12081.085, 12100.136, 12119.187, 12138.238, 12157.289, 12176.34, 12195.391, 12214.442, 12233.493, 12252.544, 12271.595, 12290.646, 12309.697, 12328.748, 12347.799, 12366.85, 12385.901, 12404.952, 12424.003, 12443.054, 12462.105, 12481.156, 12500.207, 12519.258, 12538.309, 12557.36, 12576.411, 12595.462, 12614.513, 12633.564, 12652.615, 12671.666, 12690.717, 12709.768, 12728.819, 12747.87, 12766.921, 12785.972, 12805.023, 12824.074, 12843.125, 12862.176, 12881.227, 12900.278, 12919.329, 12938.38, 12957.431, 12976.482, 12995.533, 13014.584, 13033.635, 13052.686, 13071.737, 13090.788, 13109.839, 13128.89, 13147.941, 13166.992, 13186.043, 13205.094, 13224.145, 13243.196, 13262.247, 13281.298, 13300.349, 13319.4, 13338.451, 13357.502, 13376.553, 13395.604, 13414.655, 13433.706, 13452.757, 13471.808, 13490.859, 13509.91, 13528.961, 13548.012, 13567.063, 13586.114, 13605.165, 13624.216, 13643.267, 13662.318, 13681.369, 13700.42, 13719.471, 13738.522, 13757.573, 13776.624, 13795.675, 13814.726, 13833.777, 13852.828, 13871.879, 13890.93, 13909.981, 13929.032, 13948.083, 13967.134, 13986.185, 14005.236, 14024.287, 14043.338, 14062.389, 14081.44, 14100.491, 14119.542, 14138.593, 14157.644, 14176.695, 14195.746, 14214.797, 14233.848, 14252.899, 14271.95, 14290.999, 14310.05, 14329.101, 14348.152, 14367.203, 14386.254, 14405.305, 14424.356, 14443.407, 14462.458, 14481.509, 14500.56, 14519.611, 14538.662, 14557.713, 14576.764, 14595.815, 14614.866, 14633.917, 14652.968, 14672.019, 14691.07, 14710.121, 14729.172, 14748.223, 14767.274, 14786.325, 14805.376, 14824.427, 14843.478, 14862.529, 14881.58, 14900.631, 14919.682, 14938.733, 14957.784, 14976.835, 14995.886, 15014.937, 15033.988, 15053.039, 15072.09, 15091.141, 15110.192, 15129.243, 15148.294, 15167.345, 15186.396, 15205.447, 15224.498, 15243.549, 15262.6, 15281.651, 15300.702, 15319.753, 15338.804, 15357.855, 15376.906, 15395.957, 15415.008, 15434.059, 15453.11, 15472.161, 15491.212, 15510.263, 15529.314, 15548.365, 15567.416, 15586.467, 15605.518, 15624.569, 15643.62, 15662.671, 15681.722, 15700.773, 15719.824, 15738.875, 15757.926, 15776.977, 15796.028, 15815.079, 15834.13, 15853.181, 15872.232, 15891.283, 15910.334, 15929.385, 15948.436, 15967.487, 15986.538, 16005.589, 16024.64, 16043.691, 16062.742, 16081.793, 16100.844, 16119.895, 16138.946, 16157.997, 16177.048, 16196.099, 16215.15, 16234.201, 16253.252, 16272.303, 16291.354, 16310.405, 16329.456, 16348.507, 16367.558, 16386.609, 16405.66, 16424.711, 16443.762, 16462.813, 16481.864, 16500.915, 16519.966, 16539.017, 16558.068, 16577.119, 16596.17, 16615.221, 16634.272, 16653.323, 16672.374, 16691.425, 16710.476, 16729.527, 16748.578, 16767.629, 16786.68, 16805.731, 16824.782, 16843.833, 16862.884, 16881.935, 16900.986, 16920.037, 16939.088, 16958.139, 16977.19, 16996.241, 17015.292, 17034.343, 17053.394, 17072.445, 17091.496, 17110.547, 17129.598, 17148.649, 17167.7, 17186.751, 17205.802, 17224.853, 17243.904, 17262.955, 17282.006, 17301.057, 17320.108, 17339.159, 17358.21, 17377.261, 17396.312, 17415.363, 17434.414, 17453.465, 17472.516, 17491.567, 17510.618, 17529.669, 17548.72, 17567.771, 17586.822, 17605.873, 17624.924, 17643.975, 17663.026, 17682.077, 17701.128, 17720.179, 17739.23, 17758.281, 17777.332, 17796.383, 17815.434, 17834.485, 17853.536, 17872.587, 17891.638, 17910.689, 17929.74, 17948.791, 17967.842, 17986.893, 18005.944, 18024.995, 18044.046, 18063.097, 18082.148, 18101.199, 18120.25, 18139.301, 18158.352, 18177.403, 18196.454, 18215.505, 18234.556, 18253.607, 18272.658, 18291.709, 18310.76, 18329.811, 18348.862, 18367.913, 18386.964, 18406.015, 18425.066, 18444.117, 18463.168, 18482.219, 18501.27, 18520.321, 18539.372, 18558.423, 18577.474, 18596.525, 18615.576, 18634.627, 18653.678, 18672.729, 18691.78, 18710.831, 18729.882, 18748.933, 18767.984, 18787.035, 18806.086, 18825.137, 18844.188, 18863.239, 18882.29, 18901.341, 18920.392, 18939.443, 18958.494, 18977.545, 18996.596, 19015.647, 19034.698, 19053.749, 19072.8, 19091.851, 19110.902, 19129.953, 19149.004, 19168.055, 19187.106, 19206.157, 19225.208, 19244.259, 19263.31, 19282.361, 19301.412, 19320.463, 19339.514, 19358.565, 19377.616, 19396.667, 19415.718, 19434.769, 19453.82, 19472.871, 19491.922, 19510.973, 19530.024, 19549.075, 19568.126, 19587.177, 19606.228, 19625.279, 19644.33, 19663.381, 19682.432, 19701.483, 19720.534, 19739.585, 19758.636, 19777.687, 19796.738, 19815.789, 19834.84, 19853.891, 19872.942, 19891.993, 19911.044, 19930.095, 19949.146, 19968.197, 19987.248, 20006.299, 20025.35, 20044.401, 20063.452, 20082.503, 20101.554, 20120.605, 20139.656, 20158.707, 20177.758, 20196.809, 20215.86, 20234.911, 20253.962, 20273.013, 20292.064, 20311.115, 20330.166, 20349.217, 20368.268, 20387.319, 20406.37, 20425.421, 20444.472, 20463.523, 20482.574, 20501.625, 20520.676, 20539.727, 20558.778, 20577.829, 20596.88, 20615.931, 20634.982, 20654.033, 20673.084, 20692.135, 20711.186, 20730.237, 20749.288, 20768.339, 20787.39, 20806.441, 20825.492, 20844.543, 20863.594, 20882.645, 20901.696, 20920.747, 20939.798, 20958.849, 20977.9, 20996.951, 21016.002, 21035.053, 21054.104, 21073.155, 21092.206, 21111.257, 21130.308, 21149.359, 21168.41, 21187.461, 21206.512, 21225.563, 21244.614, 21263.665, 21282.716, 21301.767, 21320.818, 21339.869, 21358.92, 21377.971, 
```

```
}
```

## 2 Model and data loading

Trained models:

```
temp = readRDS("05_final_model.rds")
fit.gg = temp$gg
fit.km0 = temp$km0
data.nswpcn = temp$data.train
```

```
data.glasgow = readRDS("06_Glasgow.rds")
data.glasgow = data.glasgow[data.glasgow$Path.Type %in% c("Pancreatic Adenocarcinoma", "Pancreatic adenocarcinoma"), ]
data.glasgow$Path.LN.Negative = data.glasgow$Path.LN.Inspected - data.glasgow$Path.LN.Involved
data.glasgow$History.Diagnosis.AgeAt = data.glasgow$History.Diagnosis.AgeAt.Cent + 68
data.glasgow$Path.Size = data.glasgow$Path.Size.Cent + 30
data.glasgow$SexM = data.glasgow$Patient.Sex == "M"
data.glasgow$AgeCent = data.glasgow$History.Diagnosis.AgeAt.Cent
data.glasgow$SizeCent = data.glasgow$Path.Size.Cent
data.glasgow$A2 = data.glasgow$Molec.S100A2.DCThresh
data.glasgow$A4 = data.glasgow$Molec.S100A4.DCThresh
data.glasgow$LocBody = data.glasgow$Path.Location != "HOP"
data.glasgow$Time = data.glasgow$History.Death.EventTimeDays
data.glasgow$DSD = data.glasgow$History.DSDeath.Event
```

```
data.apgi = readRDS("06_APGI.rds")
```

```
data.dresden = readRDS("06_Dresden.rds")

data.dresden$History.Diagnosis.AgeAt = data.dresden$History.Surgery.AgeAtYears
data.dresden$History.Diagnosis.AgeAt.Cent = data.dresden$History.Diagnosis.AgeAt - 68
data.dresden$Path.Size = data.dresden$Path.TumourSizeMm
data.dresden$Path.Size.Cent = data.dresden$Path.Size - 30
data.dresden$Stage.pT.Simplified = c("T1" = "T1", "T2" = "T2", "T3" = "T34", "T4" = "T34") [data.dresden$Stage.pT.Simplified != "T34", ]
data.dresden$Patient.Sex = data.dresden$Patient.Gender
data.dresden$SexM = data.dresden$Patient.Sex == "M"
data.dresden$AgeCent = data.dresden$History.Diagnosis.AgeAt.Cent
data.dresden$SizeCent = data.dresden$Path.Size.Cent
data.dresden$A2 = data.dresden$Molec.S100A2.DCThresh
data.dresden$A4 = data.dresden$Molec.S100A4.DCThresh
data.dresden$Path.LocationBody = data.dresden$Path.TumourLocation != "Head"
data.dresden$LocBody = data.dresden$Path.LocationBody
data.dresden$Time = data.dresden$History.Death.EventTimeDays
data.dresden$DSD = data.dresden$History.DSDeath.Event
data.dresden$Treat.MarginPositive = data.dresden$Treat.Surgery.ExcisionStatus != "R0"
data.dresden$Path.Differentiation = data.dresden$Path.Grade

temp.sel = data.dresden$Staging.pM != "M1" & !is.na(data.dresden$Staging.pM) & !is.na(data.dresden$A2) & !is.na(data.dresden$A4) & !is.na(data.dresden$LocBody) & !is.na(data.dresden$Time) & !is.na(data.dresden$DSD) & !is.na(data.dresden$Treat.MarginPositive) & !is.na(data.dresden$Path.Differentiation)
data.dresden = data.dresden[temp.sel,]
```

```
summary(data.nswpcn)
```

```
##      Patient.ID      Patient.Sex Cohort.ICGC      History.PreviousMalignancy
## Min.      :    4      F:120      Mode :logical      Mode :logical
## 1st Qu.: 305      M:120      FALSE:240      FALSE:219
## Median : 621                      NA's :0      TRUE :21
## Mean      : 618                      NA's :0
## 3rd Qu.:1030
## Max.      :1453
##
## History.FdrWithPancCancer History.FdrWithAnyCancer History.Diagnosis.Date
## Mode :logical      Mode :logical      Min.      :1994-03-09
## FALSE:230      FALSE:202      1st Qu.:1998-06-26
## TRUE :8      TRUE :38      Median :2001-05-24
## NA's :2      NA's :0      Mean      :2000-12-19
##                      3rd Qu.:2003-06-16
##                      Max.      :2006-08-14
##
## History.Diagnosis.AgeAt History.AlcoholLevel History.Smoking.Status
## Min.      :28.0      0:151      Never      :140
## 1st Qu.:62.0      1: 45      Ceased      : 48
## Median :69.0      2: 22      Current: 52
## Mean      :67.5      3: 22
## 3rd Qu.:75.0
## Max.      :87.0
##
## History.Smoking.PackYears History.Comorbid.Diabetes
## Min.      : 2.0      Mode :logical
## 1st Qu.:20.0      FALSE:181
## Median :25.0      TRUE :59
## Mean      :31.9      NA's :0
## 3rd Qu.:50.0
## Max.      :80.0
## NA's      :185
## History.Comorbid.ChronicPancreatitis History.Recurrence.Event
## Mode :logical      Min.      :0.000
## FALSE:229      1st Qu.:1.000
## TRUE :11      Median :1.000
## NA's :0      Mean      :0.971
##                      3rd Qu.:1.000
##                      Max.      :1.000
##
## History.Recurrence.Date History.DSDeath.Event History.Death.Date
## Min.      :1994-07-21      Min.      :0.000      Min.      :1995-01-12
## 1st Qu.:1999-09-16      1st Qu.:1.000      1st Qu.:1999-11-30
## Median :2002-06-03      Median :1.000      Median :2002-11-21
## Mean      :2002-03-05      Mean      :0.963      Mean      :2002-08-01
## 3rd Qu.:2005-01-08      3rd Qu.:1.000      3rd Qu.:2005-04-21
## Max.      :2009-01-29      Max.      :1.000      Max.      :2011-10-03
## NA's      :79
## History.Followup.Date History.Death.EventTimeDays Treat.Resected
## Min.      :2009-10-24      Min.      : 26      Mode:logical
## 1st Qu.:2009-10-24      1st Qu.: 274      TRUE:240
## Median :2009-10-24      Median : 476      NA's:0
```

```

## Mean :2010-01-06 Mean : 592
## 3rd Qu.:2010-02-12 3rd Qu.: 771
## Max. :2010-06-03 Max. :2701
## NA's :237
## Treat.ProcedureWhipple Treat.MarginPositive Treat.Chemo.Any
## Mode :logical Mode :logical Mode :logical
## FALSE:44 FALSE:137 FALSE:97
## TRUE :196 TRUE :103 TRUE :117
## NA's :0 NA's :0 NA's :26
##
##
##
## Treat.Chemo.Adjuvant Treat.Chemo.Adjuvant.GE3Cycles
## Mode :logical Mode :logical
## FALSE:169 FALSE:197
## TRUE :71 TRUE :43
## NA's :0 NA's :0
##
##
##
## Treat.Chemo.Palliative Treat.Chemo.PalliativeDC Treat.Chemo.GEM
## Mode :logical Mode :logical Mode :logical
## FALSE:1 FALSE:170 FALSE:151
## TRUE :65 TRUE :70 TRUE :88
## NA's :174 NA's :0 NA's :1
##
##
##
## Treat.Radio Path.LocationBody Path.Size Path.Bilirubin.Preop
## Mode :logical Mode :logical Min. : 8.0 Min. : 0.06
## FALSE:197 FALSE:196 1st Qu.:25.0 1st Qu.: 0.69
## TRUE :43 TRUE :44 Median :30.0 Median : 3.63
## NA's :0 NA's :0 Mean :33.6 Mean : 7.31
## 3rd Qu.:40.0 3rd Qu.:10.72
## Max. :90.0 Max. :45.03
## NA's :96
##
## Path.Ca199.Preop Path.Bilirubin.Postop Path.Ca199.Postop
## Min. : 1 Min. : 0.12 Min. : 1
## 1st Qu.: 73 1st Qu.: 0.47 1st Qu.: 17
## Median : 218 Median : 0.70 Median : 77
## Mean : 2803 Mean : 1.95 Mean : 1571
## 3rd Qu.: 842 3rd Qu.: 1.30 3rd Qu.: 278
## Max. :101075 Max. :25.38 Max. :31760
## NA's :162 NA's :100 NA's :137
##
## Path.Subtype Path.Differentiation Path.LN.Involved
## Adenosquamous: 18 1: 16 Min. : 0.00
## Large Cell : 0 2:157 1st Qu.: 0.00
## Mucinous : 5 3: 67 Median : 1.00
## NotSpecified : 38 4: 0 Mean : 1.76
## Papillary : 2 3rd Qu.: 2.00
## Tubular :177 Max. :12.00
## NA's :3
##
## Path.LN.Inspected Path.Invasion.Vascular Path.Invasion.Perineural
## Min. : 0.00 Mode :logical Mode :logical

```

```

## 1st Qu.: 5.00      FALSE:128      FALSE:58
## Median : 8.00      TRUE :112      TRUE :182
## Mean : 9.68      NA's :0      NA's :0
## 3rd Qu.:13.00
## Max. :52.00
## NA's :20
## Stage.pT Stage.pN Stage.pM Molec.BNIP3.NucInt Molec.BNIP3.CytoInt
## Tis: 0 NO : 80 MO :177 0 : 6 0 : 1
## T1 : 18 N1 :156 M1 : 8 1 :200 1 :125
## T2 : 32 NA's: 4 NA's: 55 2 : 21 2 : 74
## T3 :190 3 : 2 3 : 29
## T4 : 0 NA's: 11 NA's: 11
##
##
## Molec.CCND1.CytoLo Molec.CCND1.CytoHi Molec.CCND1.MembLo
## 0 :152 0 :71 0 :96
## 1 : 34 1 :87 1 :68
## 2 : 4 2 :32 2 :18
## 3 : 1 3 : 1 3 : 9
## NA's: 49 NA's:49 NA's:49
##
##
## Molec.CCND1.MembHi Molec.Grb7.Int Molec.Grb7.Percent Molec.HCNT3PlusHENT1
## 0 :29 0 :49 Min. : 0.0 Mode :logical
## 1 :86 1 :90 1st Qu.: 3.0 FALSE:93
## 2 :45 2 :42 Median : 18.0 TRUE :94
## 3 :31 3 : 7 Mean : 31.6 NA's :53
## NA's:49 NA's:52 3rd Qu.: 58.5
## Max. :100.0
## NA's :52
##
## Molec.HENT1.Percent Molec.HENT1.Int Molec.HER2 Molec.HOXB2.Percent
## Min. : 0.0 0 : 17 Mode :logical Min. : 0.0
## 1st Qu.: 11.2 1 :114 FALSE:36 1st Qu.: 35.0
## Median : 42.5 2 : 51 TRUE :10 Median : 70.0
## Mean : 44.4 3 : 12 NA's :194 Mean : 59.6
## 3rd Qu.: 75.0 NA's: 46 3rd Qu.: 85.0
## Max. :100.0 Max. :100.0
## NA's :46 NA's :42
##
## Molec.HOXB2.Int Molec.RON.Int Molec.S100A2.Int Molec.S100A2.Percent
## 0 : 14 0 : 19 0:87 Min. : 0.0
## 1 :137 1 :110 1:59 1st Qu.: 0.0
## 2 : 33 2 : 59 2:56 Median : 10.0
## 3 : 14 3 : 10 3:38 Mean : 28.1
## NA's: 42 NA's: 42 3rd Qu.: 60.0
## Max. :100.0
##
##
## Molec.S100A2.StromaScore Molec.S100A4.CytoInt Molec.S100A4.CytoPercent
## Mode :logical 0:70 Min. : 0.0
## FALSE:175 1:89 1st Qu.: 0.0
## TRUE :22 2:40 Median : 10.0
## NA's :43 3:41 Mean : 34.8
## 3rd Qu.: 75.0
## Max. :100.0
##

```

```

## Molec.S100A4.NucInt Molec.S100A4.NucPercent Stage.Overall
## 0:78 Min. : 0.0 IIB :117
## 1:66 1st Qu.: 0.0 IIA : 41
## 2:62 Median : 5.0 IB : 12
## 3:34 Mean : 26.4 IV : 8
## 3rd Qu.: 60.0 IA : 7
## Max. :100.0 (Other): 0
## NA's : 55
## History.Death.Event Molec.S100A4.DCThresh Molec.S100A2.DCThresh
## Min. :0.000 Mode :logical Mode :logical
## 1st Qu.:1.000 FALSE:60 FALSE:203
## Median :1.000 TRUE :180 TRUE :37
## Mean :0.996 NA's :0 NA's :0
## 3rd Qu.:1.000
## Max. :1.000
##
## Stage.pT.Simplified Path.Ca199.Preop.Cent Path.Ca199.Postop.Cent
## T1 : 18 Min. : -5.38 Min. : -3.97
## T2 : 32 1st Qu.: -1.09 1st Qu.: -1.14
## T34:190 Median : 0.00 Median : 0.37
## Mean : 0.09 Mean : 0.62
## 3rd Qu.: 1.36 3rd Qu.: 1.66
## Max. : 6.14 Max. : 6.40
## NA's :162 NA's :137
## History.Diagnosis.AgeAt.Cent History.Smoking.PackYears.Cent
## Min. : -40.00 Min. : -28.00
## 1st Qu.: -6.00 1st Qu.: -10.00
## Median : 1.00 Median : -5.00
## Mean : -0.51 Mean : 1.89
## 3rd Qu.: 7.00 3rd Qu.: 20.00
## Max. : 19.00 Max. : 50.00
## NA's :185
## Path.Size.Cent Path.Bilirubin.Preop.Cent Path.Bilirubin.Postop.Cent
## Min. : -22.00 Min. : -3.39 Min. : -0.53
## 1st Qu.: -5.00 1st Qu.: -2.76 1st Qu.: -0.18
## Median : 0.00 Median : 0.18 Median : 0.06
## Mean : 3.56 Mean : 3.86 Mean : 1.30
## 3rd Qu.: 10.00 3rd Qu.: 7.27 3rd Qu.: 0.66
## Max. : 60.00 Max. : 41.58 Max. : 24.74
## NA's :96 NA's :100
## History.Diagnosis.Date.Cent Path.LN.InvolvedFraction Path.LN.Negative
## Min. : -2867 Min. : 0.000 Min. : 0.00
## 1st Qu.: -1297 1st Qu.: 0.000 1st Qu.: 4.00
## Median : -234 Median : 0.143 Median : 7.00
## Mean : -389 Mean : 0.217 Mean : 7.85
## 3rd Qu.: 519 3rd Qu.: 0.333 3rd Qu.: 11.00
## Max. : 1674 Max. : 1.000 Max. : 45.00
## NA's :21 NA's :20
## SexM Ca199 DiagYearCent Time
## Mode :logical Mode :logical Min. : -7.849 Min. : 26
## FALSE:120 FALSE:26 1st Qu.: -3.551 1st Qu.: 274
## TRUE :120 TRUE :52 Median : -0.639 Median : 474
## NA's :0 NA's :162 Mean : -1.065 Mean : 589
## 3rd Qu.: 1.422 3rd Qu.: 764

```



```

##                                     Max.   : 4.583   Max.    :2701
##
##      DSD                AgeCent        LocBody        SizeCent
## Mode :logical   Min.   :-40.00   Mode :logical   Min.    :-22.00
## FALSE:9        1st Qu.: -6.00   FALSE:196       1st Qu.: -5.00
## TRUE :231       Median :  1.00   TRUE :44        Median :  0.00
## NA's :0         Mean  : -0.51   NA's :0         Mean   :  3.56
##                3rd Qu.:  7.00           3rd Qu.: 10.00
##                Max.   : 19.00           Max.    : 60.00
##
##      A2                A4                SizePlus
## Mode :logical   Mode :logical   Min.    : 0.00
## FALSE:203       FALSE:60        1st Qu.: 0.00
## TRUE :37        TRUE :180        Median : 0.00
## NA's :0         NA's :0          Mean   : 7.35
##                3rd Qu.:10.00
##                Max.    :60.00
##
summary(data.glasgow)

## Patient.ID      Patient.Sex History.Diagnosis.AgeAt Treat.Procedure
## Length:189      F: 89        Min.    :37.5          Length:189
## Class :character M:100       1st Qu.:57.8          Class :character
## Mode  :character Median :64.0          Mode  :character
##                Mean   :62.6
##                3rd Qu.:69.4
##                Max.    :86.0
##
## Path.Location
## Length:189
## Class :character
## Mode  :character
##
##
##
##
## Path.Type
## Pancreatic Adenocarcinoma :156
## Pancreatic adenocarcinoma : 32
## Pancreatic Adenocarcinom  :  1
## Pancreatic adenocarcinoma arising form IPMN :  0
## Pancreatic adenocarcinoma arising from mucnous cystic neoplsm:  0
## Pancreatic Adenocarcinoma arising IPMN :  0
## (Other) :  0
## Path.Differentiation Path.Grade Stage.pT Stage.pN
## 1: 12                Low :128 Tis:  0 NO: 33
## 2:117                High: 61 T1 :  1 N1:156
## 3: 60                T2 : 13
## 4:  0                T3 :171
##                T4 :  4
##
##
## Path.Invasion.Perineural Path.Invasion.Vascular Path.LN.Inspected

```

```

## Mode :logical          Mode :logical          Min.   : 1.0
## FALSE:13              FALSE:96                1st Qu.:13.0
## TRUE :176             TRUE :93                 Median :20.0
## NA's :0               NA's :0                 Mean    :20.2
##                                     3rd Qu.:27.0
##                                     Max.    :53.0
##
## Path.LN.InvolvedFraction Treat.MarginPositive Treat.VeinResection
## Min.   :0.00          Mode :logical          Mode :logical
## 1st Qu.:0.05          FALSE:51          FALSE:158
## Median :0.14          TRUE :138          TRUE :31
## Mean    :0.20          NA's :0           NA's :0
## 3rd Qu.:0.27
## Max.    :1.00
##
## Path.Size      History.Death.EventTimeDays History.Death.Cause
## Min.   : 5.0    Min.   : 8          0: 9
## 1st Qu.:25.0    1st Qu.: 233        1:161
## Median :30.0    Median : 501        2: 19
## Mean    :32.7    Mean    : 673
## 3rd Qu.:40.0    3rd Qu.: 915
## Max.    :65.0    Max.    :3531
##
## Treat.Chemo.Adjuvant Treat.Chemo.Neoadjuvant Molec.S100A2.DCThresh
## Mode :logical          Mode :logical          Mode :logical
## FALSE:110             FALSE:188             FALSE:127
## TRUE :79              TRUE :1               TRUE :62
## NA's :0               NA's :0               NA's :0
##
##
##
## Molec.S100A4.DCThresh Treat.ProcedureWhipple Path.LocationBody
## Mode :logical          Mode:logical          Mode :logical
## FALSE:55              TRUE:189              FALSE:189
## TRUE :134             NA's:0                NA's :0
## NA's :0
##
##
##
## History.DSDeath.Event History.ACDeath.Event Path.LN.Involved
## Mode :logical          Mode :logical          Min.   : 0.00
## FALSE:28              FALSE:9                1st Qu.: 1.00
## TRUE :161             TRUE :180              Median : 2.00
## NA's :0               NA's :0                Mean    : 3.57
##                                     3rd Qu.: 5.00
##                                     Max.    :32.00
##
##
## History.Diagnosis.AgeAt.Cent Path.Size.Cent Stage.pT.Simplified
## Min.   : -30.55          Min.   : -25.00      T1 : 1
## 1st Qu.: -10.19          1st Qu.: -5.00      T2 : 13
## Median : -4.00           Median : 0.00       T34:175
## Mean    : -5.37          Mean    : 2.72
## 3rd Qu.: 1.43           3rd Qu.: 10.00
## Max.    : 18.00          Max.    : 35.00

```

```

##
## Path.LN.Negative      SexM      AgeCent      SizeCent
## Min. : 0.0      Mode :logical  Min. : -30.55  Min. : -25.00
## 1st Qu.:10.0      FALSE:89      1st Qu.: -10.19 1st Qu.: -5.00
## Median :16.0      TRUE :100      Median : -4.00  Median : 0.00
## Mean :16.6      NA's :0        Mean : -5.37  Mean : 2.72
## 3rd Qu.:23.0      3rd Qu.: 1.43  3rd Qu.: 10.00
## Max. :47.0      Max. : 18.00  Max. : 35.00
##
##      A2      A4      LocBody      Time
## Mode :logical  Mode :logical  Mode :logical  Min. : 8
## FALSE:127      FALSE:55      FALSE:189      1st Qu.: 233
## TRUE :62       TRUE :134      NA's :0         Median : 501
## NA's :0        NA's :0         Mean : 673
##                                     3rd Qu.: 915
##                                     Max. :3531
##
##      DSD
## Mode :logical
## FALSE:28
## TRUE :161
## NA's :0
##
##
##
summary(data.apgi)

## Patient.ID      Patient.Gender      Patient.Ethnicity
## Length:75      Female:34      Asian : 7
## Class :character  Male :41      Asian, White/Caucasian : 0
## Mode :character      Black/African : 1
##                                     Black/African, White/Caucasian: 0
##                                     White/Caucasian :67
##
##
## Patient.Country History.LastFollowup.Date
## Australia :75 Min. :2008-04-14
## Italy : 0 1st Qu.:2011-02-03
## New Zealand : 0 Median :2012-05-09
## Puerto Rico : 0 Mean :2012-06-02
## United Kingdom : 0 3rd Qu.:2013-11-06
## United States of America: 0 Max. :2014-09-08
##
## History.Smoking.PackYears History.Diagnosis.Date
## Min. : 0.75 Min. :2004-12-30
## 1st Qu.:12.00 1st Qu.:2009-11-28
## Median :27.50 Median :2010-05-28
## Mean :30.98 Mean :2010-06-08
## 3rd Qu.:44.06 3rd Qu.:2010-11-29
## Max. :123.75 Max. :2012-02-17
## NA's :43
## History.Diagnosis.AgeAtYears History.Surgery.Date
## Min. :47.0 Min. :2004-12-30

```

```

## 1st Qu.:60.5          1st Qu.:2009-12-05
## Median :67.0          Median :2010-06-01
## Mean   :66.8          Mean   :2010-06-16
## 3rd Qu.:74.0          3rd Qu.:2011-01-19
## Max.   :84.0          Max.   :2012-02-17
##
##
## Treat.Surgery.Procedure
## Classic Whipple :55
## Classic Whipple, Exploratory laparotomy : 3
## PPPD : 3
## Splenectomy, Subtotal Panc/L sided Panc or distal Panc : 3
## Subtotal Panc/L sided Panc or distal Panc : 3
## Cholecystectomy, Cholecystojejunostomy/Hepaticojejunostomy, Classic Whipple: 1
## (Other) : 7
## Treat.Surgery.ExcisionStatus Treat.Surgery.Margin.Pancreatic
## R0:51 <2 mm : 2
## R1:20 Clear :65
## R2: 4 Involved: 2
## NA's : 6
##
##
## Treat.Surgery.MarginSizeMm.Pancreatic Treat.Surgery.Margin.Periunc
## Min. : 0.00 <2 mm :16
## 1st Qu.: 5.00 Clear :36
## Median :10.00 Involved:11
## Mean : 9.94 NA's :12
## 3rd Qu.:10.00
## Max. :40.00
## NA's :15
## Treat.Surgery.MarginSizeMm.Periunc Treat.Surgery.Margin.PVGroove
## Min. : 0.00 <2 mm :18
## 1st Qu.: 1.00 Clear :37
## Median : 2.20 Involved:10
## Mean : 6.92 NA's :10
## 3rd Qu.:10.00
## Max. :40.00
## NA's :24
## Treat.Surgery.MarginSizeMm.PVGroove Treat.Surgery.Margin.Retrop
## Min. : 0.0 <2 mm :19
## 1st Qu.: 1.0 Clear :46
## Median : 2.0 Involved: 5
## Mean : 3.8 NA's : 5
## 3rd Qu.: 4.0
## Max. :25.0
## NA's :24
## Treat.Surgery.MarginSizeMm.Retrop Treat.Surgery.Margin.CBD
## Min. : 0.10 <2 mm : 0
## 1st Qu.: 1.00 Clear :58
## Median : 3.00 Involved: 0
## Mean : 5.29 NA's :17
## 3rd Qu.: 8.00
## Max. :25.00
## NA's :14

```

```

## Treat.Surgery.MarginSizeMm.CBD Treat.Surgery.Margin.Duodenal
## Min. : 3.0 Clear :40
## 1st Qu.:11.5 Involved: 0
## Median :20.0 NA's :35
## Mean :21.9
## 3rd Qu.:30.0
## Max. :50.0
## NA's :31
## Treat.Surgery.MarginSizeMm.Duodenal Treat.Surgery.Margin.Gastric
## Min. : 20.0 Clear:39
## 1st Qu.: 47.5 NA's :36
## Median : 75.0
## Mean : 75.0
## 3rd Qu.:102.5
## Max. :130.0
## NA's :73
## Treat.Surgery.MarginSizeMm.Gastric Treat.Surgery.Margin.Comments
## Min. : 20 Length:75
## 1st Qu.: 40 Class :character
## Median : 60 Mode :character
## Mean : 60
## 3rd Qu.: 80
## Max. :100
## NA's :73
## Path.HistoType
## Pancreatic Ductal Adenocarcinoma:75
## Acinar Cell Carcinoma : 0
## Ampullary Adenocarcinoma : 0
## Carcinoid Tumour : 0
## Cholangiocarcinoma : 0
## Clear Cell Carcinoma : 0
## (Other) : 0
## Path.HistoType.Subtype Path.Grade
## Gastric : 0 1: 3
## Intestinal : 0 2:47
## Mixed : 0 3:23
## Not otherwise Specified (NOS):10 4: 2
## Pancreatobiliary :10
## Squamous : 0
## NA's :55
## Path.TumourLocation Path.TumourSizeMm Path.Invasion.PN
## Head :55 Min. :15.0 Absent : 9
## Head (Uncinate): 9 1st Qu.:28.0 Present:66
## Body : 7 Median :35.0
## Tail : 3 Mean :36.9
## Ampulla : 1 3rd Qu.:43.0
## : 0 Max. :90.0
## (Other) : 0
## Path.Invasion.VS Path.Nodes.Regional.Total Path.Nodes.Regional.Involved
## Absent :22 Min. : 2.0 Min. : 0.00
## Present:51 1st Qu.:13.0 1st Qu.: 1.00
## NA's : 2 Median :16.0 Median : 3.00
## Mean :18.6 Mean : 3.03
## 3rd Qu.:23.5 3rd Qu.: 4.00

```

```

##           Max.      :46.0           Max.      :13.00
##
## Path.Nodes.SepRec.Total Path.Nodes.SepRec.Involved
## Min.      : 2.0           Min.      : 0.00
## 1st Qu.:13.0           1st Qu.: 1.00
## Median :16.0           Median : 3.00
## Mean      :18.6           Mean      : 3.03
## 3rd Qu.:23.5           3rd Qu.: 4.00
## Max.      :46.0           Max.      :13.00
##
##
##           Staging.Version Staging.pM Staging.pN
## pTNM AJCC 6th Ed 2002           :12      M0 : 2      N0:16
## pTNM AJCC 7th Ed 2010           :63      M1 : 4      N1:59
## pTNM AJCC 7th Ed 2010 (Ampulla) : 0      NA's:69
## pTNM AJCC 7th Ed 2010 (Cholangiocarcinoma): 0
## pTNM AJCC 7th Ed 2010 (Neuroendocrine) : 0
##
##
## Staging.pT Staging.Stage History.Recurrence History.Recurrence.Date
## Tis: 0      IA : 1      Not observed:15      Min.      :2007-12-31
## T1 : 1      IB : 1      Suspected : 2      1st Qu.:2010-10-25
## T2 : 3      IIA:13      Confirmed :56      Median :2011-04-11
## T3 :70      IIB:55      NA's : 2      Mean :2011-06-29
## T4 : 1      III: 1      3rd Qu.:2012-02-28
##           IV : 4      Max. :2014-08-27
##           NA's :17
##
## History.Recurrence.Site.Stomach History.Recurrence.Site.Peritoneum
## Mode :logical      Mode :logical
## FALSE:75      FALSE:67
## NA's :0      TRUE :8
##           NA's :0
##
##
## History.Recurrence.Site.PancRemnant History.Recurrence.Site.PancBed
## Mode :logical      Mode :logical
## FALSE:70      FALSE:64
## TRUE :5      TRUE :11
## NA's :0      NA's :0
##
##
## History.Recurrence.Site.Other History.Recurrence.Site.Omentum
## Mode :logical      Mode :logical
## FALSE:69      FALSE:74
## TRUE :6      TRUE :1
## NA's :0      NA's :0
##
##
## History.Recurrence.Site.Mesentery History.Recurrence.Site.LymphNodes
## Mode :logical      Mode :logical
## FALSE:74      FALSE:61
## TRUE :1      TRUE :14

```

```

## NA's :0 NA's :0
##
##
##
## History.Recurrence.Site.Lung History.Recurrence.Site.Liver
## Mode :logical Mode :logical
## FALSE:60 FALSE:51
## TRUE :15 TRUE :24
## NA's :0 NA's :0
##
##
##
## History.Recurrence.Site.Brain History.Recurrence.Site.Bone
## Mode :logical Mode :logical
## FALSE:73 FALSE:71
## TRUE :2 TRUE :4
## NA's :0 NA's :0
##
##
##
## History.Status History.Death.Date
## Alive - With Disease : 7 Min. :2008-05-13
## Alive - Without Disease :13 1st Qu.:2010-12-20
## Deceased - Of Disease :51 Median :2011-12-28
## Deceased - Of Other Cause : 4 Mean :2011-11-08
## Deceased - Of Unknown Cause: 0 3rd Qu.:2012-09-08
## Max. :2014-01-26
## NA's :20
##
## History.Death.Cause Surv.Event.Death
## Cancer Death (Pancreatic) :51 Min. :0.000
## Died of Treatment Complication : 2 1st Qu.:0.000
## Cancer Death (Other) - Lung ca : 1 Median :1.000
## Other (please specify) - Suicide: 1 Mean :0.733
## Other (please specify) : 0 3rd Qu.:1.000
## (Other) : 0 Max. :1.000
## NA's :20
## Surv.EventTimeFromDiag.Death Surv.EventTimeFromSurg.Death
## Min. : 56 Min. : 62
## 1st Qu.: 386 1st Qu.: 362
## Median : 653 Median : 655
## Mean : 753 Mean : 745
## 3rd Qu.:1007 3rd Qu.:1010
## Max. :2848 Max. :2848
##
## Surv.EventTimeFromRec.Death Surv.Event.DSDeath
## Min. : 3.0 Min. :0.00
## 1st Qu.: 65.8 1st Qu.:0.00
## Median : 202.0 Median :1.00
## Mean : 287.4 Mean :0.68
## 3rd Qu.: 371.2 3rd Qu.:1.00
## Max. :1333.0 Max. :1.00
## NA's :17
## Surv.EventTimeFromDiag.DSDeath Surv.EventTimeFromSurg.DSDeath
## Min. : 31 Min. : 37

```

```

## 1st Qu.: 386          1st Qu.: 362
## Median : 653          Median : 655
## Mean   : 752          Mean    : 743
## 3rd Qu.:1007          3rd Qu.:1010
## Max.    :2848          Max.     :2848
##
## Surv.EventTimeFromRec.DSDeath Surv.Event.Recurrence
## Min.     : 3.0          Min.      :0.000
## 1st Qu.: 65.8          1st Qu.:1.000
## Median : 202.0          Median :1.000
## Mean    : 287.1          Mean     :0.767
## 3rd Qu.: 371.2          3rd Qu.:1.000
## Max.    :1333.0          Max.      :1.000
## NA's     :17            NA's      :2
## Surv.EventTimeFromDiag.Recurrence Surv.EventTimeFromSurg.Recurrence
## Min.     : 31           Min.      : -15
## 1st Qu.: 241            1st Qu.: 231
## Median : 388            Median : 377
## Mean    : 540            Mean     : 532
## 3rd Qu.: 698            3rd Qu.: 705
## Max.    :1954            Max.      :1954
## NA's     :2             NA's      :2
##      A2      A4      Path.LN.Inspected Path.LN.Involved
## Mode :logical Mode :logical Min.      : 2.0      Min.      : 0.00
## FALSE:64      FALSE:26      1st Qu.:13.0     1st Qu.: 1.00
## TRUE :11       TRUE :49       Median :16.0     Median : 3.00
## NA's :0        NA's :0        Mean   :18.6     Mean   : 3.03
##                                     3rd Qu.:23.5     3rd Qu.: 4.00
##                                     Max.    :46.0     Max.    :13.00
##
## Path.LN.Negative History.Diagnosis.AgeAt History.Diagnosis.AgeAt.Cent
## Min.      : 2.0      Min.      :47.0      Min.      : -21.00
## 1st Qu.: 9.0        1st Qu.:60.5      1st Qu.: -7.50
## Median :13.0        Median :67.0      Median : -1.00
## Mean   :15.6        Mean   :66.8      Mean   : -1.15
## 3rd Qu.:21.0        3rd Qu.:74.0      3rd Qu.: 6.00
## Max.   :44.0        Max.   :84.0      Max.   : 16.00
##
## Path.Size Path.Size.Cent Patient.Sex SexM
## Min.      :15.0      Min.      : -15.00 Female:34 Mode :logical
## 1st Qu.:28.0      1st Qu.: -2.00 Male :41  FALSE:34
## Median :35.0      Median : 5.00      TRUE :41
## Mean   :36.9      Mean   : 6.89      NA's :0
## 3rd Qu.:43.0      3rd Qu.: 13.00
## Max.   :90.0      Max.   : 60.00
##
## Treat.MarginPositive AgeCent SizeCent Stage.pT
## Mode :logical      Min.      : -21.00 Min.      : -15.00 Tis: 0
## FALSE:51           1st Qu.: -7.50 1st Qu.: -2.00 T1 : 1
## TRUE :24            Median : -1.00 Median : 5.00 T2 : 3
## NA's :0             Mean   : -1.15 Mean   : 6.89 T3 :70
##                     3rd Qu.: 6.00 3rd Qu.: 13.00 T4 : 1
##                     Max.    : 16.00 Max.    : 60.00
##

```



```

## Stage.pT.Simplified Path.LocationBody Path.Differentiation
## Length:75          Mode :logical      1: 3
## Class :character   FALSE:64          2:47
## Mode :character    TRUE :11          3:23
##                   NA's :0            4: 2
##
##
##
## LocBody           Time           DSD
## Mode :logical     Min.    : 37      Min.    :0.00
## FALSE:64          1st Qu.: 362      1st Qu.:0.00
## TRUE :11          Median : 655      Median :1.00
## NA's :0           Mean   : 743      Mean   :0.68
##                   3rd Qu.:1010      3rd Qu.:1.00
##                   Max.    :2848      Max.    :1.00
##
summary(data.dresden)

## Dresden.SSID Patient.Gender History.Surgery.AgeAtYears
## 3_105_PaCa: 1 F:68          Min.    :40.0
## 3_112_PaCa: 1 M:82          1st Qu.:59.0
## 3_11_PaCa : 1              Median :68.0
## 3_131_PaCa: 1              Mean   :65.6
## 3_13_PaCa : 1              3rd Qu.:73.0
## 3_196_PaCa: 1              Max.    :84.0
## (Other) :144
## History.Death.EventTimeDays History.Death.Event History.DSDeath.Event
## Min.    : 10              Mode :logical      Mode :logical
## 1st Qu.: 311              FALSE:22          FALSE:38
## Median : 514              TRUE :128          TRUE :112
## Mean   : 715              NA's :0            NA's :0
## 3rd Qu.: 915
## Max.    :4190
##
## History.Death.Cause Treat.Surgery.ExcisionStatus Path.Grade Staging.pT
## other: 16          R0:98              1: 3      T2: 9
## PaCa :112          R1:42              2:75      T3:141
## NA's : 22          R2:10              3:71
##                   4: 1
##
##
##
## Staging.pN Staging.pM Path.Invasion.VS Path.Invasion.PN
## N0: 47      M0:150      Mode :logical      Mode :logical
## N1:101      M1: 0       FALSE:64          FALSE:53
## N2: 2                TRUE :36          TRUE :95
##                   NA's :50          NA's :2
##
##
##
## Path.TumourLocation Path.TumourSizeMm Molec.S100A2.DCThresh
## Head:139            Min.    :15.0      Mode :logical
## Tail: 11            1st Qu.:25.0      FALSE:112

```

```

##           Median :34.5      TRUE :38
##           Mean    :34.2      NA's :0
##           3rd Qu.:40.0
##           Max.    :85.0
##
## Molec.S100A4.DCThresh History.Diagnosis.AgeAt
## Mode :logical      Min.    :40.0
## FALSE:18           1st Qu.:59.0
## TRUE :132           Median :68.0
## NA's :0             Mean    :65.6
##                     3rd Qu.:73.0
##                     Max.    :84.0
##
## History.Diagnosis.AgeAt.Cent  Path.Size  Path.Size.Cent
## Min.    :-28.00              Min.    :15.0  Min.    :-15.00
## 1st Qu.: -9.00              1st Qu.:25.0  1st Qu.: -5.00
## Median :  0.00              Median :34.5  Median :  4.50
## Mean    : -2.39              Mean    :34.2  Mean     :  4.17
## 3rd Qu.:  5.00              3rd Qu.:40.0  3rd Qu.: 10.00
## Max.    : 16.00              Max.    :85.0  Max.     : 55.00
##
## Stage.pT.Simplified Patient.Sex  SexM      AgeCent
## Length:150      F:68      Mode :logical  Min.    :-28.00
## Class :character M:82      FALSE:68      1st Qu.: -9.00
## Mode  :character TRUE :82      Median :  0.00
##                     NA's :0      Mean    : -2.39
##                     3rd Qu.:  5.00
##                     Max.    : 16.00
##
##           SizeCent      A2      A4      Path.LocationBody
## Min.    :-15.00  Mode :logical  Mode :logical  Mode :logical
## 1st Qu.: -5.00  FALSE:112  FALSE:18      FALSE:139
## Median :  4.50  TRUE :38      TRUE :132      TRUE :11
## Mean    :  4.17  NA's :0      NA's :0      NA's :0
## 3rd Qu.: 10.00
## Max.    : 55.00
##
## LocBody      Time      DSD      Treat.MarginPositive
## Mode :logical  Min.    : 10  Mode :logical  Mode :logical
## FALSE:139      1st Qu.: 311  FALSE:38      FALSE:98
## TRUE :11        Median : 514  TRUE :112      TRUE :52
## NA's :0          Mean    : 715  NA's :0      NA's :0
##                 3rd Qu.: 915
##                 Max.    :4190
##
## Path.Differentiation
## 1: 3
## 2:75
## 3:71
## 4: 1
##
##
##

```

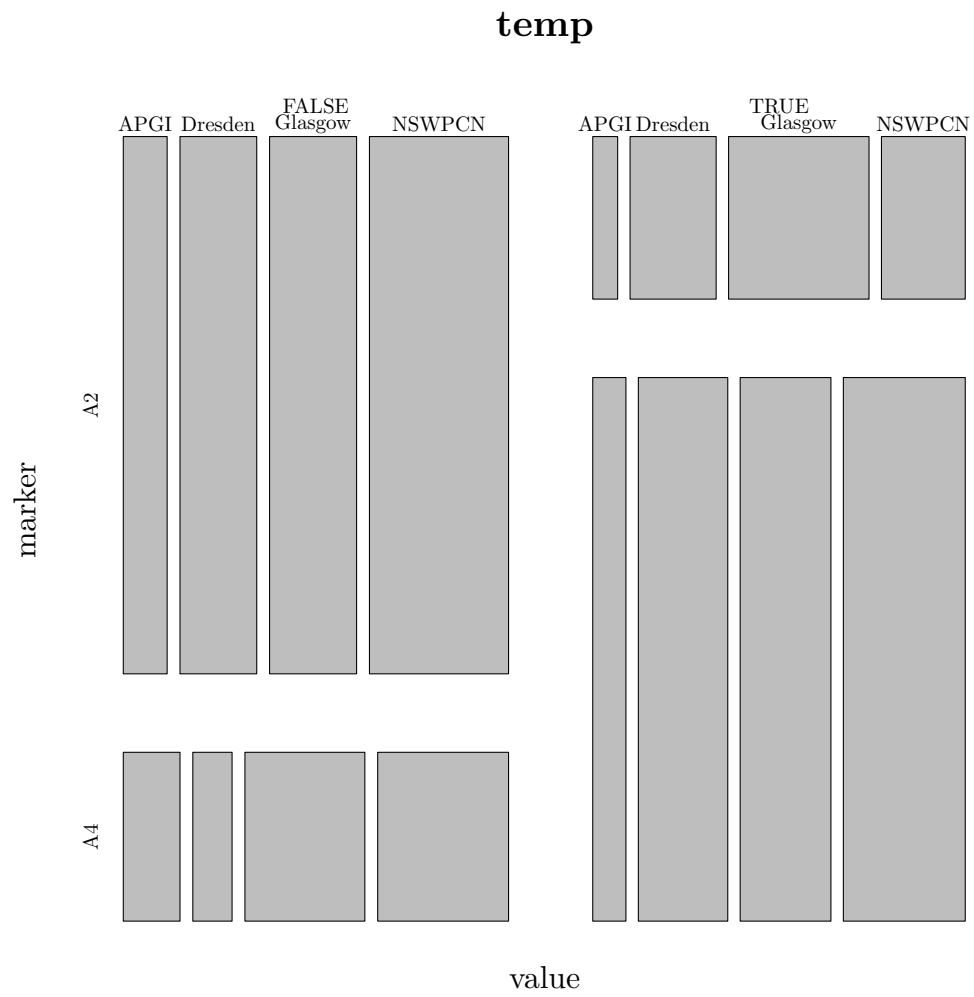
```

temp = table(value = c(data.nswpcn$A2, data.glasgow$A2, data.apgi$A2, data.dresden$A2, data.nswpcn$A4, c
temp

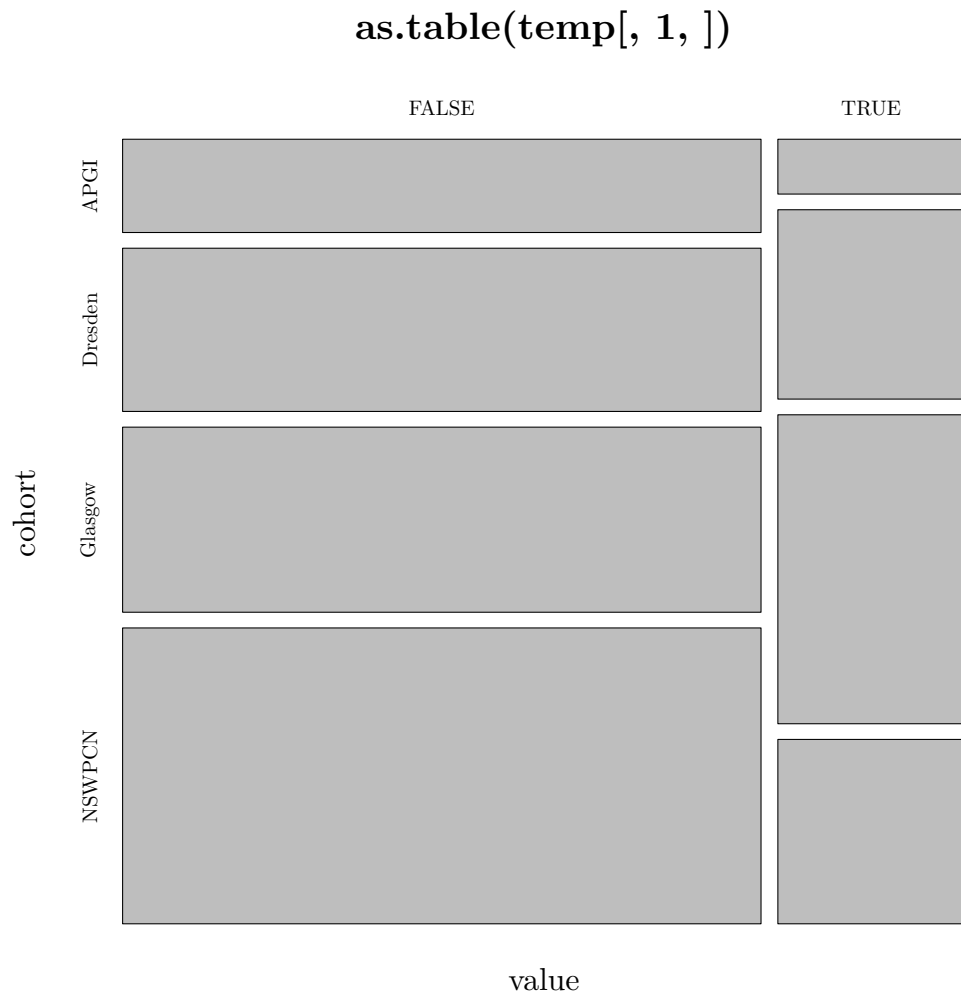
## , , cohort = APGI
##
##      marker
## value    A2  A4
## FALSE   64  26
## TRUE    11  49
##
## , , cohort = Dresden
##
##      marker
## value    A2  A4
## FALSE  112  18
## TRUE    38 132
##
## , , cohort = Glasgow
##
##      marker
## value    A2  A4
## FALSE  127  55
## TRUE    62 134
##
## , , cohort = NSWPCN
##
##      marker
## value    A2  A4
## FALSE  203  60
## TRUE    37 180

plot(temp)

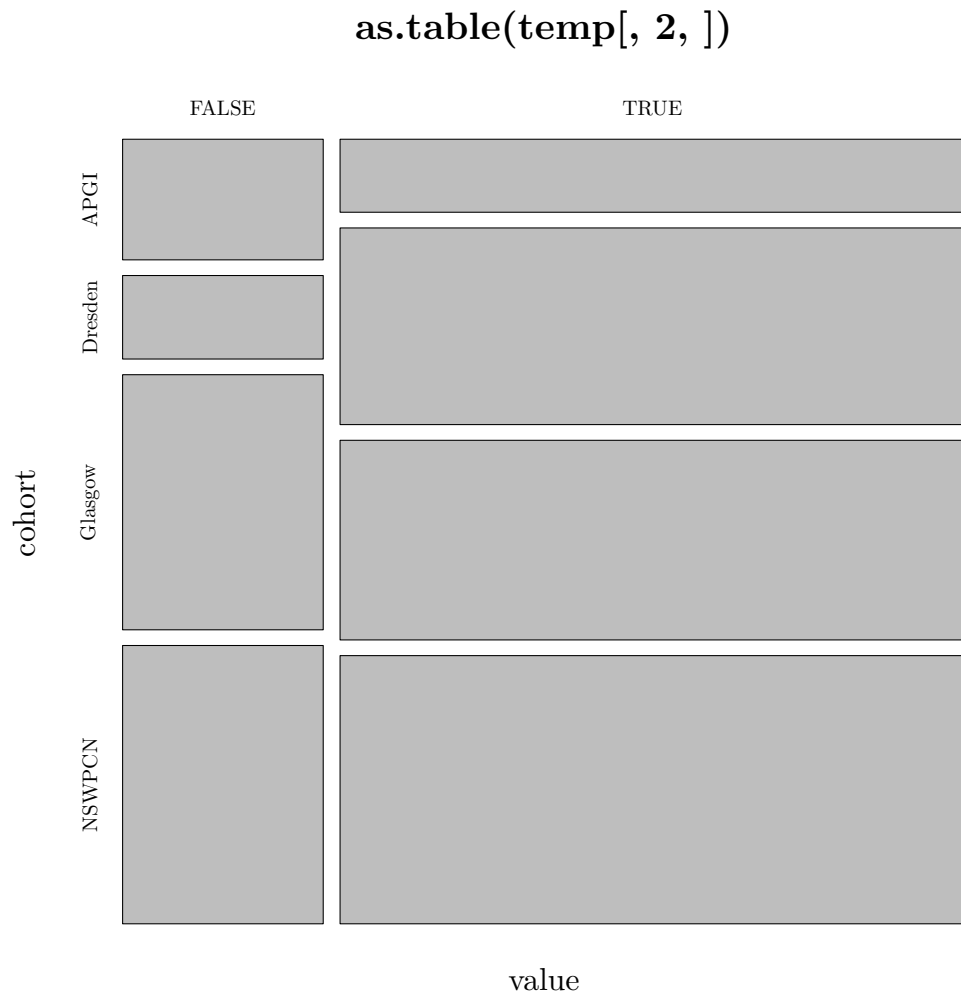
```



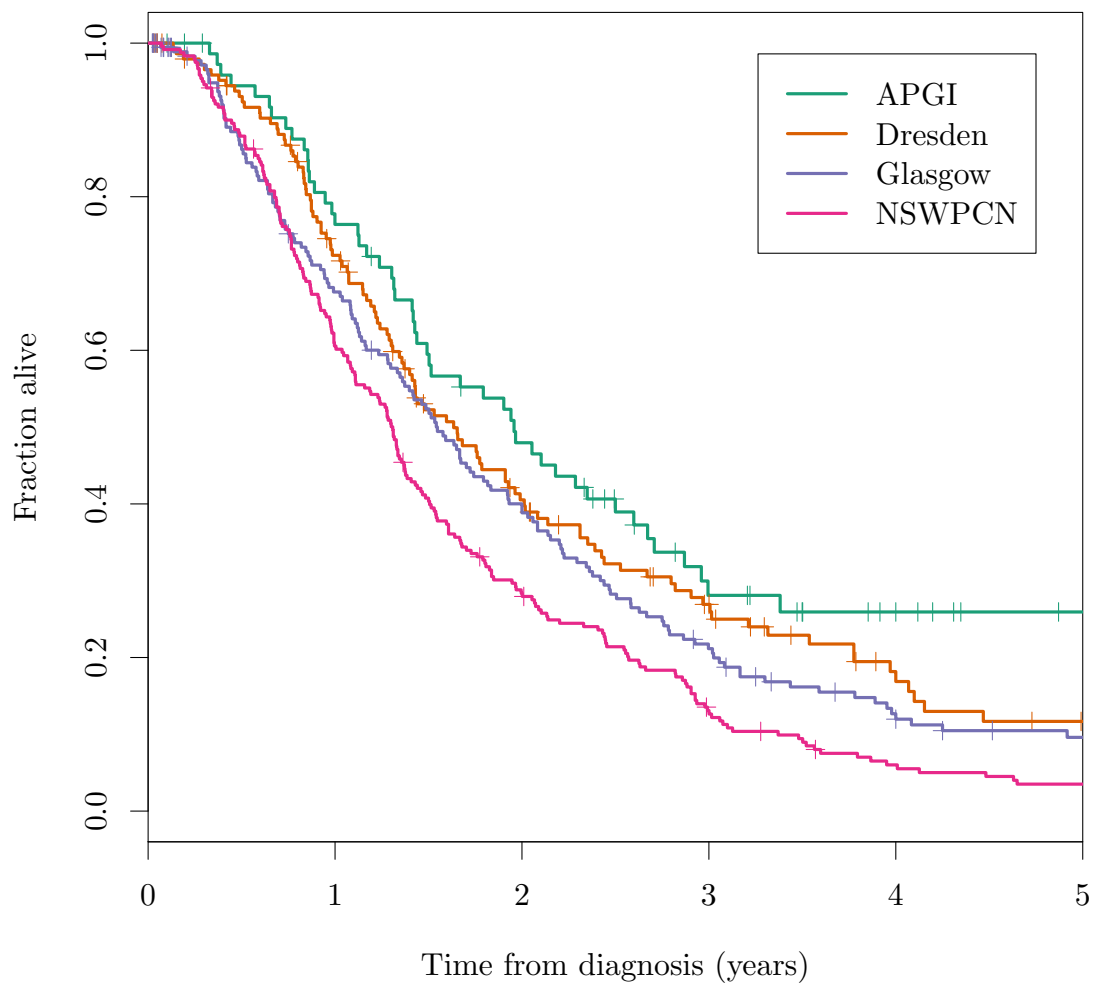
```
plot(as.table(temp[,1,]))
```



```
plot(as.table(temp[,2,]))
```



```
temp.time = c(data.nswpcn$Time, data.glasgow$Time, data.apgi$Time, data.dresden$Time) / 365.25
temp.dsd = c(data.nswpcn$DSD, data.glasgow$DSD, data.apgi$DSD, data.dresden$DSD)
temp.cohort = factor(rep(c("NSWPCN", "Glasgow", "APGI", "Dresden"), c(nrow(data.nswpcn), nrow(data.glasgow), nrow(data.apgi), nrow(data.dresden))), c("NSWPCN", "Glasgow", "APGI", "Dresden"))
temp.survfit = survfit(Surv(temp.time, temp.dsd) ~ temp.cohort)
plot(temp.survfit, col = pal[1:4], xlim = c(0, 5), lwd = 3, main = "", xlab = "Time from diagnosis (years)", ylab = "Survival probability", legend = "topright", legend = c("APGI", "Dresden", "Glasgow", "NSWPCN"), col = pal[1:4], inset = 0.05, lwd = 3)
```



```
survdif(Surv(temp.time, temp.dsd) ~ temp.cohort)

## Call:
## survdif(formula = Surv(temp.time, temp.dsd) ~ temp.cohort)
##
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## temp.cohort=APGI      75      51    74.7     7.495     8.7035
## temp.cohort=Dresden  150     112   137.3     4.661     6.2829
## temp.cohort=Glasgow  189     161   163.2     0.031     0.0443
## temp.cohort=NSWPCN  240     231   179.8    14.581    21.8988
##
##  Chisq= 27.1  on 3 degrees of freedom, p= 5.68e-06

temp.vars = c("Time", "DSD", "SexM", "AgeCent", "SizeCent", "Stage.pT.Simplified", "LocBody", "Treat.Mar
temp.all = as.data.frame(rbind(
  cbind(data.nswpcn[,temp.vars], cohort = "NSWPCN"),
  cbind(data.glasgow[,temp.vars], cohort = "Glasgow"),
  cbind(data.apgi[,temp.vars], cohort = "APGI"),
  cbind(data.dresden[,temp.vars], cohort = "Dresden")))
table(temp.all$SexM, temp.all$cohort)

##
##      NSWPCN Glasgow APGI Dresden
```

```

## FALSE      120      89   34    68
## TRUE       120     100   41    82

temp.allfit = coxph(Surv(Time, DSD) ~ LocBody + cohort*(SexM + AgeCent + SizeCent + Treat.MarginPositive)
temp.allfit2 = coxph(Surv(Time, DSD) ~ LocBody + SexM + AgeCent + SizeCent + Treat.MarginPositive + I(Pa
summary(temp.allfit)

## Call:
## coxph(formula = Surv(Time, DSD) ~ LocBody + cohort * (SexM +
##      AgeCent + SizeCent + Treat.MarginPositive + I(Path.Differentiation %in%
##      c("3", "4"))) + A2 + A4), data = temp.all)
##
##      n= 654, number of events= 555
##
##
##                                     coef
## LocBodyTRUE                        2.83e-01
## cohortGlasgow                     -8.22e-01
## cohortAPGI                        -1.08e+00
## cohortDresden                     -6.85e-01
## SexMTRUE                          -9.81e-02
## AgeCent                          -3.10e-03
## SizeCent                          8.51e-03
## Treat.MarginPositiveTRUE          4.91e-01
## I(Path.Differentiation %in% c("3", "4"))TRUE 1.35e-01
## A2TRUE                            6.88e-01
## A4TRUE                            5.70e-01
## cohortGlasgow:SexMTRUE            2.53e-01
## cohortAPGI:SexMTRUE              5.60e-01
## cohortDresden:SexMTRUE           3.91e-01
## cohortGlasgow:AgeCent             -2.27e-02
## cohortAPGI:AgeCent               2.78e-02
## cohortDresden:AgeCent            1.83e-02
## cohortGlasgow:SizeCent            2.44e-02
## cohortAPGI:SizeCent              8.54e-05
## cohortDresden:SizeCent           -4.15e-04
## cohortGlasgow:Treat.MarginPositiveTRUE 1.61e-01
## cohortAPGI:Treat.MarginPositiveTRUE 8.27e-01
## cohortDresden:Treat.MarginPositiveTRUE 7.85e-02
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE 1.74e-01
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE 2.71e-01
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE 3.62e-01
## cohortGlasgow:A2TRUE             5.56e-02
## cohortAPGI:A2TRUE                -1.99e-01
## cohortDresden:A2TRUE             -4.45e-01
## cohortGlasgow:A4TRUE             -2.69e-01
## cohortAPGI:A4TRUE               -2.65e-01
## cohortDresden:A4TRUE            -3.09e-01
##
##                                     exp(coef)
## LocBodyTRUE                        1.33e+00
## cohortGlasgow                      4.40e-01
## cohortAPGI                        3.41e-01
## cohortDresden                     5.04e-01
## SexMTRUE                          9.07e-01
## AgeCent                          9.97e-01

```



## SizeCent	1.01e+00	
## Treat.MarginPositiveTRUE	1.63e+00	
## I(Path.Differentiation %in% c("3", "4"))TRUE	1.14e+00	
## A2TRUE	1.99e+00	
## A4TRUE	1.77e+00	
## cohortGlasgow:SexMTRUE	1.29e+00	
## cohortAPGI:SexMTRUE	1.75e+00	
## cohortDresden:SexMTRUE	1.48e+00	
## cohortGlasgow:AgeCent	9.78e-01	
## cohortAPGI:AgeCent	1.03e+00	
## cohortDresden:AgeCent	1.02e+00	
## cohortGlasgow:SizeCent	1.02e+00	
## cohortAPGI:SizeCent	1.00e+00	
## cohortDresden:SizeCent	1.00e+00	
## cohortGlasgow:Treat.MarginPositiveTRUE	1.17e+00	
## cohortAPGI:Treat.MarginPositiveTRUE	2.29e+00	
## cohortDresden:Treat.MarginPositiveTRUE	1.08e+00	
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE	1.19e+00	
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE	1.31e+00	
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE	1.44e+00	
## cohortGlasgow:A2TRUE	1.06e+00	
## cohortAPGI:A2TRUE	8.19e-01	
## cohortDresden:A2TRUE	6.41e-01	
## cohortGlasgow:A4TRUE	7.64e-01	
## cohortAPGI:A4TRUE	7.67e-01	
## cohortDresden:A4TRUE	7.34e-01	
##	se(coef)	z
## LocBodyTRUE	1.64e-01	1.73
## cohortGlasgow	2.87e-01	-2.86
## cohortAPGI	3.59e-01	-2.99
## cohortDresden	3.91e-01	-1.75
## SexMTRUE	1.33e-01	-0.74
## AgeCent	7.30e-03	-0.42
## SizeCent	5.13e-03	1.66
## Treat.MarginPositiveTRUE	1.41e-01	3.48
## I(Path.Differentiation %in% c("3", "4"))TRUE	1.61e-01	0.84
## A2TRUE	2.03e-01	3.40
## A4TRUE	1.59e-01	3.59
## cohortGlasgow:SexMTRUE	2.13e-01	1.18
## cohortAPGI:SexMTRUE	3.41e-01	1.64
## cohortDresden:SexMTRUE	2.38e-01	1.64
## cohortGlasgow:AgeCent	1.12e-02	-2.03
## cohortAPGI:AgeCent	1.94e-02	1.43
## cohortDresden:AgeCent	1.32e-02	1.38
## cohortGlasgow:SizeCent	9.38e-03	2.60
## cohortAPGI:SizeCent	1.01e-02	0.01
## cohortDresden:SizeCent	1.07e-02	-0.04
## cohortGlasgow:Treat.MarginPositiveTRUE	2.38e-01	0.68
## cohortAPGI:Treat.MarginPositiveTRUE	3.38e-01	2.45
## cohortDresden:Treat.MarginPositiveTRUE	2.53e-01	0.31
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE	2.38e-01	0.73
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE	3.41e-01	0.79
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE	2.65e-01	1.36
## cohortGlasgow:A2TRUE	2.71e-01	0.21

## cohortAPGI:A2TRUE	4.36e-01	-0.46
## cohortDresden:A2TRUE	3.07e-01	-1.45
## cohortGlasgow:A4TRUE	2.42e-01	-1.11
## cohortAPGI:A4TRUE	3.73e-01	-0.71
## cohortDresden:A4TRUE	3.64e-01	-0.85
##	Pr(> z )	
## LocBodyTRUE	0.08445	
## cohortGlasgow	0.00422	
## cohortAPGI	0.00275	
## cohortDresden	0.08017	
## SexMTRUE	0.46098	
## AgeCent	0.67098	
## SizeCent	0.09729	
## Treat.MarginPositiveTRUE	0.00050	
## I(Path.Differentiation %in% c("3", "4"))TRUE	0.40055	
## A2TRUE	0.00069	
## A4TRUE	0.00034	
## cohortGlasgow:SexMTRUE	0.23670	
## cohortAPGI:SexMTRUE	0.10066	
## cohortDresden:SexMTRUE	0.10095	
## cohortGlasgow:AgeCent	0.04235	
## cohortAPGI:AgeCent	0.15134	
## cohortDresden:AgeCent	0.16648	
## cohortGlasgow:SizeCent	0.00928	
## cohortAPGI:SizeCent	0.99325	
## cohortDresden:SizeCent	0.96910	
## cohortGlasgow:Treat.MarginPositiveTRUE	0.49879	
## cohortAPGI:Treat.MarginPositiveTRUE	0.01431	
## cohortDresden:Treat.MarginPositiveTRUE	0.75612	
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE	0.46494	
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE	0.42668	
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE	0.17251	
## cohortGlasgow:A2TRUE	0.83723	
## cohortAPGI:A2TRUE	0.64777	
## cohortDresden:A2TRUE	0.14658	
## cohortGlasgow:A4TRUE	0.26585	
## cohortAPGI:A4TRUE	0.47722	
## cohortDresden:A4TRUE	0.39663	
##		
##	exp(coef)	
## LocBodyTRUE	1.327	
## cohortGlasgow	0.440	
## cohortAPGI	0.341	
## cohortDresden	0.504	
## SexMTRUE	0.907	
## AgeCent	0.997	
## SizeCent	1.009	
## Treat.MarginPositiveTRUE	1.633	
## I(Path.Differentiation %in% c("3", "4"))TRUE	1.145	
## A2TRUE	1.989	
## A4TRUE	1.768	
## cohortGlasgow:SexMTRUE	1.287	
## cohortAPGI:SexMTRUE	1.751	
## cohortDresden:SexMTRUE	1.478	

## cohortGlasgow:AgeCent	0.978
## cohortAPGI:AgeCent	1.028
## cohortDresden:AgeCent	1.018
## cohortGlasgow:SizeCent	1.025
## cohortAPGI:SizeCent	1.000
## cohortDresden:SizeCent	1.000
## cohortGlasgow:Treat.MarginPositiveTRUE	1.175
## cohortAPGI:Treat.MarginPositiveTRUE	2.287
## cohortDresden:Treat.MarginPositiveTRUE	1.082
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE	1.190
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE	1.311
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE	1.436
## cohortGlasgow:A2TRUE	1.057
## cohortAPGI:A2TRUE	0.819
## cohortDresden:A2TRUE	0.641
## cohortGlasgow:A4TRUE	0.764
## cohortAPGI:A4TRUE	0.767
## cohortDresden:A4TRUE	0.734
##	exp(-coef)
## LocBodyTRUE	0.754
## cohortGlasgow	2.275
## cohortAPGI	2.933
## cohortDresden	1.983
## SexMTRUE	1.103
## AgeCent	1.003
## SizeCent	0.992
## Treat.MarginPositiveTRUE	0.612
## I(Path.Differentiation %in% c("3", "4"))TRUE	0.874
## A2TRUE	0.503
## A4TRUE	0.566
## cohortGlasgow:SexMTRUE	0.777
## cohortAPGI:SexMTRUE	0.571
## cohortDresden:SexMTRUE	0.676
## cohortGlasgow:AgeCent	1.023
## cohortAPGI:AgeCent	0.973
## cohortDresden:AgeCent	0.982
## cohortGlasgow:SizeCent	0.976
## cohortAPGI:SizeCent	1.000
## cohortDresden:SizeCent	1.000
## cohortGlasgow:Treat.MarginPositiveTRUE	0.851
## cohortAPGI:Treat.MarginPositiveTRUE	0.437
## cohortDresden:Treat.MarginPositiveTRUE	0.924
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE	0.840
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE	0.763
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE	0.696
## cohortGlasgow:A2TRUE	0.946
## cohortAPGI:A2TRUE	1.220
## cohortDresden:A2TRUE	1.560
## cohortGlasgow:A4TRUE	1.309
## cohortAPGI:A4TRUE	1.303
## cohortDresden:A4TRUE	1.362
##	lower .95
## LocBodyTRUE	0.962
## cohortGlasgow	0.250

## cohortAPGI	0.169
## cohortDresden	0.234
## SexMTRUE	0.698
## AgeCent	0.983
## SizeCent	0.998
## Treat.MarginPositiveTRUE	1.239
## I(Path.Differentiation %in% c("3", "4"))TRUE	0.835
## A2TRUE	1.337
## A4TRUE	1.295
## cohortGlasgow:SexMTRUE	0.847
## cohortAPGI:SexMTRUE	0.897
## cohortDresden:SexMTRUE	0.927
## cohortGlasgow:AgeCent	0.956
## cohortAPGI:AgeCent	0.990
## cohortDresden:AgeCent	0.992
## cohortGlasgow:SizeCent	1.006
## cohortAPGI:SizeCent	0.980
## cohortDresden:SizeCent	0.979
## cohortGlasgow:Treat.MarginPositiveTRUE	0.737
## cohortAPGI:Treat.MarginPositiveTRUE	1.180
## cohortDresden:Treat.MarginPositiveTRUE	0.659
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE	0.746
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE	0.672
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE	0.854
## cohortGlasgow:A2TRUE	0.622
## cohortAPGI:A2TRUE	0.349
## cohortDresden:A2TRUE	0.351
## cohortGlasgow:A4TRUE	0.476
## cohortAPGI:A4TRUE	0.370
## cohortDresden:A4TRUE	0.359
##	upper .95
## LocBodyTRUE	1.830
## cohortGlasgow	0.772
## cohortAPGI	0.689
## cohortDresden	1.086
## SexMTRUE	1.177
## AgeCent	1.011
## SizeCent	1.019
## Treat.MarginPositiveTRUE	2.153
## I(Path.Differentiation %in% c("3", "4"))TRUE	1.568
## A2TRUE	2.958
## A4TRUE	2.414
## cohortGlasgow:SexMTRUE	1.956
## cohortAPGI:SexMTRUE	3.419
## cohortDresden:SexMTRUE	2.359
## cohortGlasgow:AgeCent	0.999
## cohortAPGI:AgeCent	1.068
## cohortDresden:AgeCent	1.045
## cohortGlasgow:SizeCent	1.044
## cohortAPGI:SizeCent	1.020
## cohortDresden:SizeCent	1.021
## cohortGlasgow:Treat.MarginPositiveTRUE	1.872
## cohortAPGI:Treat.MarginPositiveTRUE	4.433
## cohortDresden:Treat.MarginPositiveTRUE	1.776

```

## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE      1.897
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE        2.558
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE      2.416
## cohortGlasgow:A2TRUE                                           1.798
## cohortAPGI:A2TRUE                                              1.925
## cohortDresden:A2TRUE                                           1.169
## cohortGlasgow:A4TRUE                                           1.227
## cohortAPGI:A4TRUE                                              1.593
## cohortDresden:A4TRUE                                           1.500
##
## Concordance= 0.681 (se = 0.014 )
## Rsquare= 0.274 (max possible= 1 )
## Likelihood ratio test= 209 on 32 df, p=0
## Wald test = 202 on 32 df, p=0
## Score (logrank) test = 218 on 32 df, p=0

anova(temp.allfit)

## Analysis of Deviance Table
## Cox model: response is Surv(Time, DSD)
## Terms added sequentially (first to last)
##
##
## loglik Chisq Df Pr(>|Chi|)
## NULL -3089
## LocBody -3082 14.72 1 0.00013
## cohort -3069 26.55 3 7.3e-06
## SexM -3068 1.22 1 0.27016
## AgeCent -3068 0.15 1 0.70036
## SizeCent -3054 27.65 1 1.5e-07
## Treat.MarginPositive -3037 33.69 1 6.5e-09
## I(Path.Differentiation %in% c("3", "4")) -3026 22.66 1 1.9e-06
## A2 -3011 29.43 1 5.8e-08
## A4 -3003 17.05 1 3.6e-05
## cohort:SexM -3000 5.32 3 0.14956
## cohort:AgeCent -2994 12.21 3 0.00671
## cohort:SizeCent -2990 7.07 3 0.06978
## cohort:Treat.MarginPositive -2987 5.63 3 0.13114
## cohort:I(Path.Differentiation %in% c("3", "4")) -2987 1.06 3 0.78686
## cohort:A2 -2985 3.33 3 0.34326
## cohort:A4 -2984 1.67 3 0.64338

anova(temp.allfit, temp.allfit2)

## Analysis of Deviance Table
## Cox model: response is Surv(Time, DSD)
## Model 1: ~ LocBody + cohort * (SexM + AgeCent + SizeCent + Treat.MarginPositive + I(Path.Differentiation %in%
## Model 2: ~ LocBody + SexM + AgeCent + SizeCent + Treat.MarginPositive + I(Path.Differentiation %in%
## loglik Chisq Df P(>|Chi|)
## 1 -2984
## 2 -3025 81.3 24 3.8e-08

cox.zph(temp.allfit)

## rho
## LocBodyTRUE 0.03809

```

## cohortGlasgow	0.06697
## cohortAPGI	0.01437
## cohortDresden	-0.01345
## SexMTRUE	0.11035
## AgeCent	-0.07038
## SizeCent	-0.05970
## Treat.MarginPositiveTRUE	-0.05022
## I(Path.Differentiation %in% c("3", "4"))TRUE	-0.00879
## A2TRUE	0.02464
## A4TRUE	-0.06118
## cohortGlasgow:SexMTRUE	-0.07548
## cohortAPGI:SexMTRUE	-0.10355
## cohortDresden:SexMTRUE	-0.05615
## cohortGlasgow:AgeCent	0.03511
## cohortAPGI:AgeCent	-0.06395
## cohortDresden:AgeCent	0.00267
## cohortGlasgow:SizeCent	-0.01654
## cohortAPGI:SizeCent	0.02391
## cohortDresden:SizeCent	-0.00991
## cohortGlasgow:Treat.MarginPositiveTRUE	0.00379
## cohortAPGI:Treat.MarginPositiveTRUE	0.06396
## cohortDresden:Treat.MarginPositiveTRUE	0.01213
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE	-0.04754
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE	-0.00593
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE	0.06682
## cohortGlasgow:A2TRUE	-0.04746
## cohortAPGI:A2TRUE	0.06970
## cohortDresden:A2TRUE	-0.02126
## cohortGlasgow:A4TRUE	-0.02756
## cohortAPGI:A4TRUE	0.01480
## cohortDresden:A4TRUE	-0.01282
## GLOBAL	NA
##	chisq
## LocBodyTRUE	0.90616
## cohortGlasgow	2.56256
## cohortAPGI	0.10117
## cohortDresden	0.12785
## SexMTRUE	6.70881
## AgeCent	3.28661
## SizeCent	2.35545
## Treat.MarginPositiveTRUE	1.50171
## I(Path.Differentiation %in% c("3", "4"))TRUE	0.04807
## A2TRUE	0.38624
## A4TRUE	2.05228
## cohortGlasgow:SexMTRUE	3.23367
## cohortAPGI:SexMTRUE	6.67723
## cohortDresden:SexMTRUE	1.91757
## cohortGlasgow:AgeCent	0.71607
## cohortAPGI:AgeCent	2.76090
## cohortDresden:AgeCent	0.00445
## cohortGlasgow:SizeCent	0.18172
## cohortAPGI:SizeCent	0.35473
## cohortDresden:SizeCent	0.07646
## cohortGlasgow:Treat.MarginPositiveTRUE	0.00837

```

## cohortAPGI:Treat.MarginPositiveTRUE 2.63651
## cohortDresden:Treat.MarginPositiveTRUE 0.09593
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE 1.29374
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE 0.02202
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE 2.94642
## cohortGlasgow:A2TRUE 1.29339
## cohortAPGI:A2TRUE 2.82302
## cohortDresden:A2TRUE 0.28947
## cohortGlasgow:A4TRUE 0.40662
## cohortAPGI:A4TRUE 0.13340
## cohortDresden:A4TRUE 0.10976
## GLOBAL 58.78746
##
## P
## LocBodyTRUE 0.34114
## cohortGlasgow 0.10942
## cohortAPGI 0.75043
## cohortDresden 0.72067
## SexMTRUE 0.00959
## AgeCent 0.06985
## SizeCent 0.12485
## Treat.MarginPositiveTRUE 0.22041
## I(Path.Differentiation %in% c("3", "4"))TRUE 0.82645
## A2TRUE 0.53428
## A4TRUE 0.15198
## cohortGlasgow:SexMTRUE 0.07214
## cohortAPGI:SexMTRUE 0.00977
## cohortDresden:SexMTRUE 0.16613
## cohortGlasgow:AgeCent 0.39744
## cohortAPGI:AgeCent 0.09659
## cohortDresden:AgeCent 0.94683
## cohortGlasgow:SizeCent 0.66990
## cohortAPGI:SizeCent 0.55145
## cohortDresden:SizeCent 0.78216
## cohortGlasgow:Treat.MarginPositiveTRUE 0.92710
## cohortAPGI:Treat.MarginPositiveTRUE 0.10443
## cohortDresden:Treat.MarginPositiveTRUE 0.75677
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE 0.25536
## cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE 0.88203
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE 0.08607
## cohortGlasgow:A2TRUE 0.25542
## cohortAPGI:A2TRUE 0.09292
## cohortDresden:A2TRUE 0.59056
## cohortGlasgow:A4TRUE 0.52369
## cohortAPGI:A4TRUE 0.71493
## cohortDresden:A4TRUE 0.74042
## GLOBAL 0.00267

temp = cox.zph(temp.allfit)$table
sort(p.adjust(temp[grepl("^cohort", rownames(temp)), "p"], "holm"))

## cohortAPGI:SexMTRUE
## 0.2344
## cohortGlasgow
## 1.0000

```

```

##                                cohortAPGI
##                                1.0000
##                                cohortDresden
##                                1.0000
##                                cohortGlasgow:SexMTRUE
##                                1.0000
##                                cohortDresden:SexMTRUE
##                                1.0000
##                                cohortGlasgow:AgeCent
##                                1.0000
##                                cohortAPGI:AgeCent
##                                1.0000
##                                cohortDresden:AgeCent
##                                1.0000
##                                cohortGlasgow:SizeCent
##                                1.0000
##                                cohortAPGI:SizeCent
##                                1.0000
##                                cohortDresden:SizeCent
##                                1.0000
##                                cohortGlasgow:Treat.MarginPositiveTRUE
##                                1.0000
##                                cohortAPGI:Treat.MarginPositiveTRUE
##                                1.0000
##                                cohortDresden:Treat.MarginPositiveTRUE
##                                1.0000
## cohortGlasgow:I(Path.Differentiation %in% c("3", "4"))TRUE
##                                1.0000
##      cohortAPGI:I(Path.Differentiation %in% c("3", "4"))TRUE
##                                1.0000
## cohortDresden:I(Path.Differentiation %in% c("3", "4"))TRUE
##                                1.0000
##                                cohortGlasgow:A2TRUE
##                                1.0000
##                                cohortAPGI:A2TRUE
##                                1.0000
##                                cohortDresden:A2TRUE
##                                1.0000
##                                cohortGlasgow:A4TRUE
##                                1.0000
##                                cohortAPGI:A4TRUE
##                                1.0000
##                                cohortDresden:A4TRUE
##                                1.0000

#plot(cox.zph(temp.allfit))

```

### 3 Score calculation

```
temp = applyNomogram(fit.mskcc, data.glasgow)
```



```
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Portal.Vein
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Splenectomy
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Posterior.Margin
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Back.pain
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Weight.loss

mskcc_post.linpred.glasgow = temp[,1]
mskcc_post.12mo.glasgow = temp[,2]
mskcc_post.24mo.glasgow = temp[,3]
mskcc_post.36mo.glasgow = temp[,4]
temp = applyNomogram(fit.mskcc, data.glasgow[,c("History.Diagnosis.AgeAt", "Patient.Sex", "Path.Location", "Path.Differentiation", "Path.LN.Involved", "Path.LN.Negative", "Treat.MarginPositive", "Posterior.Margin", "Back.pain", "Weight.loss", "Portal.Vein", "Splenectomy")], data.glasgow[,c("History.Diagnosis.AgeAt", "Patient.Sex", "Path.Location", "Path.Differentiation", "Path.LN.Involved", "Path.LN.Negative", "Treat.MarginPositive", "Posterior.Margin", "Back.pain", "Weight.loss", "Portal.Vein", "Splenectomy")])

## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Portal.Vein
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Splenectomy
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Treat.MarginPositive
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.Differentiation
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Posterior.Margin
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.LN.Involved
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.LN.Negative
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Back.pain
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Weight.loss

mskcc_pre.linpred.glasgow = temp[,1]
mskcc_pre.12mo.glasgow = temp[,2]
mskcc_pre.24mo.glasgow = temp[,3]
mskcc_pre.36mo.glasgow = temp[,4]
```

```
temp = applyNomogram(fit.mskcc, data.apgi)

## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Portal.Vein
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Splenectomy
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Posterior.Margin
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Back.pain
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Weight.loss

mskcc_post.linpred.apgi = temp[,1]
```

```

mskcc_post.12mo.apgi = temp[,2]
mskcc_post.24mo.apgi = temp[,3]
mskcc_post.36mo.apgi = temp[,4]
temp = applyNomogram(fit.mskcc, data.apgi[,c("History.Diagnosis.AgeAt", "Patient.Sex", "Path.LocationBoo

## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Portal.Vein
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Splenectomy
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Treat.MarginPositive
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.Differentiation
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Posterior.Margin
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.LN.Involved
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.LN.Negative
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Back.pain
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Weight.loss

mskcc_pre.linpred.apgi = temp[,1]
mskcc_pre.12mo.apgi = temp[,2]
mskcc_pre.24mo.apgi = temp[,3]
mskcc_pre.36mo.apgi = temp[,4]

```

```

temp = applyNomogram(fit.mskcc, data.dresden)

## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Portal.Vein
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Splenectomy
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Posterior.Margin
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.LN.Involved
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.LN.Negative
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Back.pain
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Weight.loss

mskcc_post.linpred.dresden = temp[,1]
mskcc_post.12mo.dresden = temp[,2]
mskcc_post.24mo.dresden = temp[,3]
mskcc_post.36mo.dresden = temp[,4]
temp = applyNomogram(fit.mskcc, data.dresden[,c("History.Diagnosis.AgeAt", "Patient.Sex", "Path.LocationBoo

## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Portal.Vein

```

```
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Splenectomy
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Treat.MarginPositive
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.Differentiation
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Posterior.Margin
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.LN.Involved
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Path.LN.Negative
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Back.pain
## Warning in FUN(c("History.Diagnosis.AgeAt", "Patient.Sex", "Portal.Vein", : Marginalizing
missing variable: Weight.loss

mskcc_pre.linpred.dresden = temp[,1]
mskcc_pre.12mo.dresden = temp[,2]
mskcc_pre.24mo.dresden = temp[,3]
mskcc_pre.36mo.dresden = temp[,4]
```

Get approximate linear predictors from the GG model, by just calculating the location term.

```
val.prob.times = seq(0, max(c(data.glasgow$Time, data.apgi$Time)), 1)
```

```
gg.path.glasgow = summary(fit.gg, newdata = data.glasgow, ci = FALSE)
temp.coefs = coef(fit.gg)
gg.linpred.glasgow = sapply(1:length(temp.coefs), function(coef_i) {
  # if (names(temp.coefs)[coef_i] == "SexMTRUE") {
  #   rep(0, nrow(data.val))
  # } else
  if (names(temp.coefs)[coef_i] %in% colnames(data.glasgow)) {
    temp.coefs[coef_i] * data.glasgow[,names(temp.coefs)[coef_i]]
  } else if (gsub("TRUE$", "", names(temp.coefs)[coef_i]) %in% colnames(data.glasgow)) {
    temp.coefs[coef_i] * data.glasgow[,gsub("TRUE$", "", names(temp.coefs)[coef_i])]
  } else {
    rep(0, nrow(data.glasgow))
  } })
gg.linpred.glasgow = -rowSums(gg.linpred.glasgow) # Negate to bring into concordance with the dir
temp = summary(fit.gg, newdata = data.glasgow, ci = FALSE)
gg.prob.glasgow = sapply(temp, function(x) approx(x[,1], x[,2], xout = val.prob.times, yleft = 1, yright = 0))
colnames(gg.prob.glasgow) = rownames(data.glasgow)
```

```
gg.path.apgi = summary(fit.gg, newdata = data.apgi, ci = FALSE)
temp.coefs = coef(fit.gg)
gg.linpred.apgi = sapply(1:length(temp.coefs), function(coef_i) {
  # if (names(temp.coefs)[coef_i] == "SexMTRUE") {
  #   rep(0, nrow(data.val))
  # } else
  if (names(temp.coefs)[coef_i] %in% colnames(data.apgi)) {
    temp.coefs[coef_i] * data.apgi[,names(temp.coefs)[coef_i]]
  } else {
    rep(0, nrow(data.apgi))
  } })
```

```

    } else if (gsub("TRUE$", "", names(temp.coefs)[coef_i]) %in% colnames(data.apgi)) {
      temp.coefs[coef_i] * data.apgi[,gsub("TRUE$", "", names(temp.coefs)[coef_i])]
    } else {
      rep(0, nrow(data.apgi))
    } })
gg.linpred.apgi = -rowSums(gg.linpred.apgi)      # Negate to bring into concordance with the direction of
temp = summary(fit.gg, newdata = data.apgi, ci = FALSE)
gg.prob.apgi = sapply(temp, function(x) approx(x[,1], x[,2], xout = val.prob.times, yleft = 1, yright =
colnames(gg.prob.apgi) = rownames(data.apgi))

```

```

gg.path.dresden = summary(fit.gg, newdata = data.dresden, ci = FALSE)
temp.coefs = coef(fit.gg)
gg.linpred.dresden = sapply(1:length(temp.coefs), function(coef_i) {
  # if (names(temp.coefs)[coef_i] == "SexMTRUE") {
  #   rep(0, nrow(data.val))
  # } else
  if (names(temp.coefs)[coef_i] %in% colnames(data.dresden)) {
    temp.coefs[coef_i] * data.dresden[,names(temp.coefs)[coef_i]]
  } else if (gsub("TRUE$", "", names(temp.coefs)[coef_i]) %in% colnames(data.dresden)) {
    temp.coefs[coef_i] * data.dresden[,gsub("TRUE$", "", names(temp.coefs)[coef_i])]
  } else {
    rep(0, nrow(data.dresden))
  } })
gg.linpred.dresden = -rowSums(gg.linpred.dresden)      # Negate to bring into concordance with the direction of
temp = summary(fit.gg, newdata = data.dresden, ci = FALSE)
gg.prob.dresden = sapply(temp, function(x) approx(x[,1], x[,2], xout = val.prob.times, yleft = 1, yright =
colnames(gg.prob.dresden) = rownames(data.dresden))

```

```

gg.linpred.nswpcn = sapply(1:length(temp.coefs), function(coef_i) {
  # if (names(temp.coefs)[coef_i] == "SexMTRUE") {
  #   rep(0, nrow(data.val))
  # } else
  if (names(temp.coefs)[coef_i] %in% colnames(data.glasgow)) {
    temp.coefs[coef_i] * data.nswpcn[,names(temp.coefs)[coef_i]]
  } else if (gsub("TRUE$", "", names(temp.coefs)[coef_i]) %in% colnames(data.nswpcn)) {
    temp.coefs[coef_i] * data.nswpcn[,gsub("TRUE$", "", names(temp.coefs)[coef_i])]
  } else {
    rep(0, nrow(data.nswpcn))
  } })
gg.linpred.nswpcn = -rowSums(gg.linpred.nswpcn)      # Negate to bring into concordance with the direction of
temp = summary(fit.gg, newdata = data.nswpcn, ci = FALSE)
gg.prob.nswpcn = sapply(temp, function(x) approx(x[,1], x[,2], xout = val.prob.times, yleft = 1, yright =
colnames(gg.prob.nswpcn) = rownames(data.nswpcn))

```

## 4 Validation

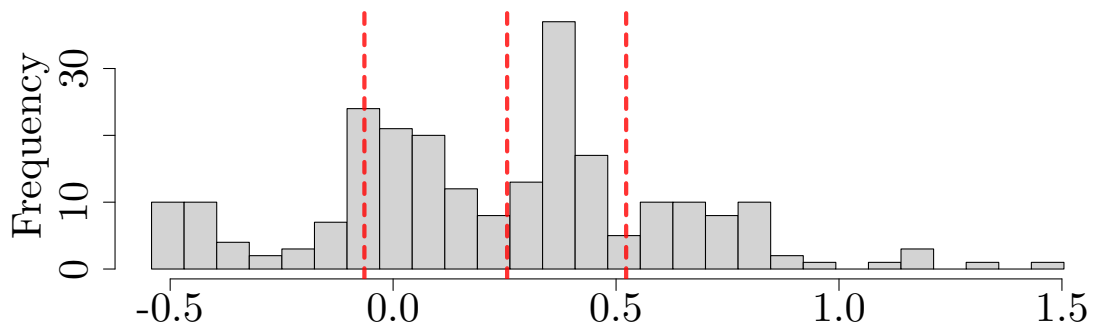
### 4.1 Altman diagnostic 1: score histograms

```

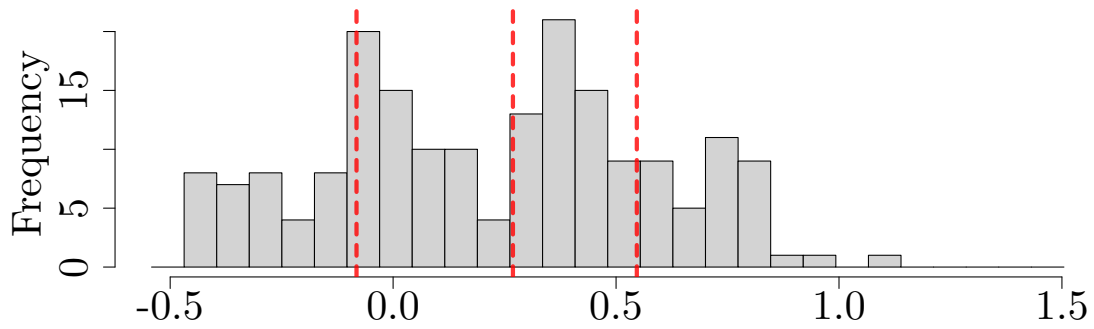
pars = par(no.readonly = TRUE)
par(mfrow = c(4, 1), mar = c(5.1, 5.1, 4.1, 2.1))
temp.breaks = seq(min(c(gg.linpred.nswpcn, gg.linpred.glasgow, gg.linpred.apgi, gg.linpred.dresden)), ma
hist(gg.linpred.nswpcn, main = "NSWPCN", xlim = range(c(gg.linpred.nswpcn, gg.linpred.glasgow, gg.linpre
abline(v = quantile(gg.linpred.nswpcn, probs = c(0.2, 0.5, 0.8)), col = rgb(1, 0, 0, 0.8), lwd = 4, lty
hist(gg.linpred.glasgow, main = "Glasgow", xlim = range(c(gg.linpred.nswpcn, gg.linpred.glasgow, gg.linpre
abline(v = quantile(gg.linpred.glasgow, probs = c(0.2, 0.5, 0.8)), col = rgb(1, 0, 0, 0.8), lwd = 4, lty
hist(gg.linpred.apgi, main = "APGI", xlim = range(c(gg.linpred.nswpcn, gg.linpred.glasgow, gg.linpred.ap
abline(v = quantile(gg.linpred.apgi, probs = c(0.2, 0.5, 0.8)), col = rgb(1, 0, 0, 0.8), lwd = 4, lty =
hist(gg.linpred.dresden, main = "Dresden", xlim = range(c(gg.linpred.nswpcn, gg.linpred.glasgow, gg.linpre
abline(v = quantile(gg.linpred.dresden, probs = c(0.2, 0.5, 0.8)), col = rgb(1, 0, 0, 0.8), lwd = 4, lty

```

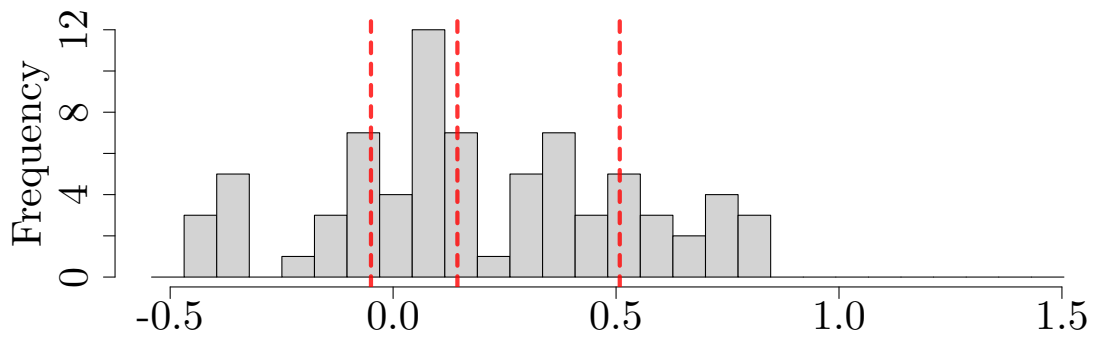
**NSWPCN**



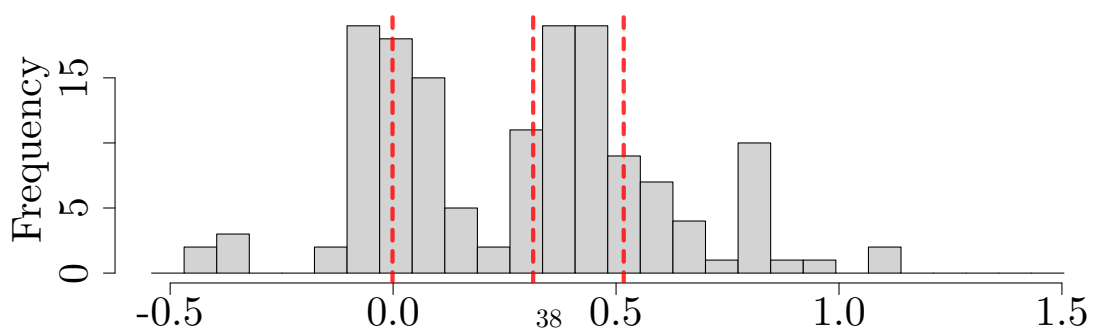
**Glasgow**



**APGI**



**Dresden**



```
par(pars)
```

## 4.2 Altman method 1 (D,F)

```
summary(coxph(Surv(Time, DSD) ~ mskcc_post.linpred.glasgow, data.glasgow))

## Call:
## coxph(formula = Surv(Time, DSD) ~ mskcc_post.linpred.glasgow,
##       data = data.glasgow)
##
##      n= 189, number of events= 161
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## mskcc_post.linpred.glasgow 0.01682   1.01696  0.00428 3.93 8.4e-05
##
##               exp(coef) exp(-coef) lower .95 upper .95
## mskcc_post.linpred.glasgow      1.02      0.983      1.01      1.03
##
## Concordance= 0.584 (se = 0.026 )
## Rsquare= 0.081 (max possible= 0.999 )
## Likelihood ratio test= 15.9 on 1 df,  p=6.79e-05
## Wald test               = 15.5 on 1 df,  p=8.43e-05
## Score (logrank) test = 15.7 on 1 df,  p=7.56e-05

summary(coxph(Surv(Time, DSD) ~ mskcc_pre.linpred.glasgow, data.glasgow))

## Call:
## coxph(formula = Surv(Time, DSD) ~ mskcc_pre.linpred.glasgow,
##       data = data.glasgow)
##
##      n= 189, number of events= 161
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## mskcc_pre.linpred.glasgow 0.0118   1.0118  0.0105 1.12 0.26
##
##               exp(coef) exp(-coef) lower .95 upper .95
## mskcc_pre.linpred.glasgow      1.01      0.988      0.991      1.03
##
## Concordance= 0.585 (se = 0.026 )
## Rsquare= 0.006 (max possible= 0.999 )
## Likelihood ratio test= 1.15 on 1 df,  p=0.284
## Wald test              = 1.25 on 1 df,  p=0.263
## Score (logrank) test = 1.25 on 1 df,  p=0.264

summary(coxph(Surv(Time, DSD) ~ mskcc_post.linpred.apgi, data.apgi))

## Call:
## coxph(formula = Surv(Time, DSD) ~ mskcc_post.linpred.apgi, data = data.apgi)
##
##      n= 75, number of events= 51
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## mskcc_post.linpred.apgi 0.01626   1.01639  0.00452 3.6 0.00032
```

```
##
##               exp(coef) exp(-coef) lower .95 upper .95
## mskcc_post.linpred.apgi      1.02      0.984      1.01      1.03
##
## Concordance= 0.701 (se = 0.044 )
## Rsquare= 0.14 (max possible= 0.993 )
## Likelihood ratio test= 11.3 on 1 df, p=0.000754
## Wald test = 12.9 on 1 df, p=0.000319
## Score (logrank) test = 13.3 on 1 df, p=0.000268

summary(coxph(Surv(Time, DSD) ~ mskcc_pre.linpred.apgi, data.apgi))

## Call:
## coxph(formula = Surv(Time, DSD) ~ mskcc_pre.linpred.apgi, data = data.apgi)
##
## n= 75, number of events= 51
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## mskcc_pre.linpred.apgi 0.00329  1.00330  0.00673 0.49      0.62
##
##               exp(coef) exp(-coef) lower .95 upper .95
## mskcc_pre.linpred.apgi      1      0.997      0.99      1.02
##
## Concordance= 0.475 (se = 0.044 )
## Rsquare= 0.003 (max possible= 0.993 )
## Likelihood ratio test= 0.23 on 1 df, p=0.634
## Wald test = 0.24 on 1 df, p=0.625
## Score (logrank) test = 0.24 on 1 df, p=0.624

summary(coxph(Surv(Time, DSD) ~ mskcc_post.linpred.dresden, data.dresden))

## Call:
## coxph(formula = Surv(Time, DSD) ~ mskcc_post.linpred.dresden,
## data = data.dresden)
##
## n= 150, number of events= 112
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## mskcc_post.linpred.dresden 0.00792  1.00795  0.00363 2.18      0.029
##
##               exp(coef) exp(-coef) lower .95 upper .95
## mskcc_post.linpred.dresden      1.01      0.992      1      1.02
##
## Concordance= 0.597 (se = 0.031 )
## Rsquare= 0.028 (max possible= 0.998 )
## Likelihood ratio test= 4.2 on 1 df, p=0.0404
## Wald test = 4.76 on 1 df, p=0.0291
## Score (logrank) test = 4.81 on 1 df, p=0.0282

summary(coxph(Surv(Time, DSD) ~ mskcc_pre.linpred.dresden, data.dresden))

## Call:
## coxph(formula = Surv(Time, DSD) ~ mskcc_pre.linpred.dresden,
## data = data.dresden)
##
```



```

##      n= 150, number of events= 112
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## mskcc_pre.linpred.dresden 0.00336   1.00337  0.00485 0.69    0.49
##
##              exp(coef) exp(-coef) lower .95 upper .95
## mskcc_pre.linpred.dresden      1      0.997    0.994    1.01
##
## Concordance= 0.518 (se = 0.031 )
## Rsquare= 0.003 (max possible= 0.998 )
## Likelihood ratio test= 0.45 on 1 df,  p=0.502
## Wald test            = 0.48 on 1 df,  p=0.488
## Score (logrank) test = 0.48 on 1 df,  p=0.488

summary(coxph(Surv(Time, DSD) ~ gg.linpred.glasgow, data.glasgow))

## Call:
## coxph(formula = Surv(Time, DSD) ~ gg.linpred.glasgow, data = data.glasgow)
##
##      n= 189, number of events= 161
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## gg.linpred.glasgow 0.805      2.236   0.239 3.37  0.00075
##
##              exp(coef) exp(-coef) lower .95 upper .95
## gg.linpred.glasgow      2.24      0.447    1.4    3.57
##
## Concordance= 0.607 (se = 0.026 )
## Rsquare= 0.059 (max possible= 0.999 )
## Likelihood ratio test= 11.4 on 1 df,  p=0.000725
## Wald test            = 11.3 on 1 df,  p=0.000754
## Score (logrank) test = 11.5 on 1 df,  p=0.000705

summary(coxph(Surv(Time, DSD) ~ gg.linpred.apgi, data.apgi))

## Call:
## coxph(formula = Surv(Time, DSD) ~ gg.linpred.apgi, data = data.apgi)
##
##      n= 75, number of events= 51
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## gg.linpred.apgi 0.894      2.444   0.427 2.09   0.036
##
##              exp(coef) exp(-coef) lower .95 upper .95
## gg.linpred.apgi      2.44      0.409    1.06    5.64
##
## Concordance= 0.579 (se = 0.044 )
## Rsquare= 0.057 (max possible= 0.993 )
## Likelihood ratio test= 4.42 on 1 df,  p=0.0355
## Wald test            = 4.39 on 1 df,  p=0.0362
## Score (logrank) test = 4.43 on 1 df,  p=0.0352

summary(coxph(Surv(Time, DSD) ~ gg.linpred.dresden, data.dresden))

## Call:

```

```
## coxph(formula = Surv(Time, DSD) ~ gg.linpred.dresden, data = data.dresden)
##
##      n= 150, number of events= 112
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## gg.linpred.dresden 0.527      1.694    0.312 1.69    0.091
##
##              exp(coef) exp(-coef) lower .95 upper .95
## gg.linpred.dresden      1.69      0.59    0.919    3.12
##
## Concordance= 0.545 (se = 0.031 )
## Rsquare= 0.019 (max possible= 0.998 )
## Likelihood ratio test= 2.82 on 1 df,  p=0.0928
## Wald test               = 2.85 on 1 df,  p=0.0913
## Score (logrank) test = 2.86 on 1 df,  p=0.0911

anova(coxph(Surv(Time, DSD) ~ offset(gg.linpred.glasgow) + gg.linpred.glasgow, data.glasgow))

## Analysis of Deviance Table
## Cox model: response is Surv(Time, DSD)
## Terms added sequentially (first to last)
##
##              loglik Chisq Df Pr(>|Chi|)
## NULL                  -678
## gg.linpred.glasgow    -678  0.66  1      0.41

anova(coxph(Surv(Time, DSD) ~ offset(gg.linpred.apgi) + gg.linpred.apgi, data.apgi))

## Analysis of Deviance Table
## Cox model: response is Surv(Time, DSD)
## Terms added sequentially (first to last)
##
##              loglik Chisq Df Pr(>|Chi|)
## NULL                  -185
## gg.linpred.apgi      -185  0.06  1      0.8

anova(coxph(Surv(Time, DSD) ~ offset(gg.linpred.dresden) + gg.linpred.dresden, data.dresden))

## Analysis of Deviance Table
## Cox model: response is Surv(Time, DSD)
## Terms added sequentially (first to last)
##
##              loglik Chisq Df Pr(>|Chi|)
## NULL                  -466
## gg.linpred.dresden    -465  2.31  1      0.13
```

Booyah.

### 4.3 Altman method 2 (F)

```
summary(coxph(Surv(Time, DSD) ~ offset(mskcc_pre.linpred.glasgow) + AgeCent + SexM + SizeCent + A2 + A4,
data = data.mscc)

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, : Ran out of
iterations and did not converge
```

```
## Error in fitter(X, Y, strats, offset, init, control, weights = weights, : NA/NaN/Inf in foreign function call (arg 6)
```

```
summary(coxph(Surv(Time, DSD) ~ offset(mskcc_post.linpred.glasgow) + AgeCent + SexM + SizeCent + A2 + A4
```

```
## Call:
```

```
## coxph(formula = Surv(Time, DSD) ~ offset(mskcc_post.linpred.glasgow) +  
## AgeCent + SexM + SizeCent + A2 + A4, data = data.glasgow)
```

```
##
```

```
## n= 189, number of events= 161
```

```
##
```

	coef	exp(coef)	se(coef)	z	Pr(> z )
## AgeCent	0.22744	1.25538	0.00862	26.39	< 2e-16
## SexMTRUE	-4.18282	0.01526	0.29544	-14.16	< 2e-16
## SizeCent	0.07140	1.07401	0.01910	3.74	0.00019
## A2TRUE	-2.96537	0.05154	0.41042	-7.23	5e-13
## A4TRUE	5.40464	222.43685	0.28361	19.06	< 2e-16

```
##
```

	exp(coef)	exp(-coef)	lower .95	upper .95
## AgeCent	1.2554	0.7966	1.23e+00	1.2768
## SexMTRUE	0.0153	65.5506	8.55e-03	0.0272
## SizeCent	1.0740	0.9311	1.03e+00	1.1150
## A2TRUE	0.0515	19.4019	2.31e-02	0.1152
## A4TRUE	222.4369	0.0045	1.28e+02	387.8075

```
##
```

```
## Concordance= 0.588 (se = 0.026 )
```

```
## Rsquare= 0.982 (max possible= 1 )
```

```
## Likelihood ratio test= 757 on 5 df, p=0
```

```
## Wald test = 1654 on 5 df, p=0
```

```
## Score (logrank) test = 1745 on 5 df, p=0
```

```
summary(coxph(Surv(Time, DSD) ~ offset(gg.linpred.glasgow) + AgeCent + SexM + SizeCent + A2 + A4, data.g
```

```
## Call:
```

```
## coxph(formula = Surv(Time, DSD) ~ offset(gg.linpred.glasgow) +  
## AgeCent + SexM + SizeCent + A2 + A4, data = data.glasgow)
```

```
##
```

```
## n= 189, number of events= 161
```

```
##
```

	coef	exp(coef)	se(coef)	z	Pr(> z )
## AgeCent	-0.03105	0.96943	0.00872	-3.56	0.00037
## SexMTRUE	0.63117	1.87981	0.16671	3.79	0.00015
## SizeCent	0.02245	1.02270	0.00767	2.93	0.00343
## A2TRUE	0.33327	1.39553	0.17564	1.90	0.05776
## A4TRUE	-0.05074	0.95052	0.18482	-0.27	0.78367

```
##
```

	exp(coef)	exp(-coef)	lower .95	upper .95
## AgeCent	0.969	1.032	0.953	0.986
## SexMTRUE	1.880	0.532	1.356	2.606
## SizeCent	1.023	0.978	1.007	1.038
## A2TRUE	1.396	0.717	0.989	1.969
## A4TRUE	0.951	1.052	0.662	1.365

```
##
```

```
## Concordance= 0.676 (se = 0.026 )
```

```
## Rsquare= 0.184 (max possible= 0.999 )
```

```

## Likelihood ratio test= 38.4 on 5 df, p=3.19e-07
## Wald test = 39 on 5 df, p=2.4e-07
## Score (logrank) test = 40.5 on 5 df, p=1.19e-07

summary(coxph(Surv(Time, DSD) ~ offset(mskcc_pre.linpred.apgi) + AgeCent + SexM + SizeCent + A2 + A4, da

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, : Ran out of
iterations and did not converge
## Error in fitter(X, Y, strats, offset, init, control, weights = weights, : NA/NaN/Inf in
foreign function call (arg 6)

summary(coxph(Surv(Time, DSD) ~ offset(mskcc_post.linpred.apgi) + AgeCent + SexM + SizeCent + A2 + A4, c

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, : Ran out of
iterations and did not converge
## Error in fitter(X, Y, strats, offset, init, control, weights = weights, : NA/NaN/Inf in
foreign function call (arg 6)

summary(coxph(Surv(Time, DSD) ~ offset(gg.linpred.apgi) + AgeCent + SexM + SizeCent + A2 + A4, data.apgi)

## Call:
## coxph(formula = Surv(Time, DSD) ~ offset(gg.linpred.apgi) + AgeCent +
## SexM + SizeCent + A2 + A4, data = data.apgi)
##
## n= 75, number of events= 51
##
##      coef exp(coef) se(coef)      z Pr(>|z|)
## AgeCent  0.02404   1.02433  0.01798  1.34  0.1812
## SexMTRUE  0.99912   2.71590  0.31918  3.13  0.0017
## SizeCent  0.01343   1.01352  0.00828  1.62  0.1050
## A2TRUE    0.22816   1.25628  0.39709  0.57  0.5656
## A4TRUE    0.17023   1.18558  0.33812  0.50  0.6146
##
##      exp(coef) exp(-coef) lower .95 upper .95
## AgeCent      1.02      0.976   0.989   1.06
## SexMTRUE      2.72      0.368   1.453   5.08
## SizeCent      1.01      0.987   0.997   1.03
## A2TRUE        1.26      0.796   0.577   2.74
## A4TRUE        1.19      0.843   0.611   2.30
##
## Concordance= 0.684 (se = 0.044 )
## Rsquare= 0.189 (max possible= 0.993 )
## Likelihood ratio test= 15.7 on 5 df, p=0.00775
## Wald test = 14.8 on 5 df, p=0.0113
## Score (logrank) test = 15.6 on 5 df, p=0.00816

summary(coxph(Surv(Time, DSD) ~ offset(mskcc_pre.linpred.dresden) + AgeCent + SexM + SizeCent + A2 + A4, da

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, : Ran out of
iterations and did not converge

## Call:
## coxph(formula = Surv(Time, DSD) ~ offset(mskcc_pre.linpred.dresden) +
## AgeCent + SexM + SizeCent + A2 + A4, data = data.dresden)
##
## n= 150, number of events= 112

```

```
##
##          coef exp(coef) se(coef)      z Pr(>|z|)
## AgeCent   1.07e+00  2.90e+00  1.40e+00  0.76   0.446
## SexMTRUE -9.61e+00  6.72e-05  7.27e+00 -1.32   0.186
## SizeCent -7.39e-02  9.29e-01  3.99e-01 -0.19   0.853
## A2TRUE    9.40e-01  2.56e+00  1.18e+01  0.08   0.936
## A4TRUE    2.69e+01  4.83e+11  1.38e+01  1.95   0.052
##
##          exp(coef) exp(-coef) lower .95 upper .95
## AgeCent   2.90e+00   3.45e-01  1.87e-01  4.51e+01
## SexMTRUE   6.72e-05   1.49e+04  4.38e-11  1.03e+02
## SizeCent   9.29e-01   1.08e+00  4.25e-01  2.03e+00
## A2TRUE     2.56e+00   3.91e-01  2.42e-10  2.71e+10
## A4TRUE     4.83e+11   2.07e-12  8.14e-01  2.86e+23
##
## Concordance= 0.551 (se = 0.031 )
## Rsquare= 1 (max possible= 1 )
## Likelihood ratio test= 6039 on 5 df, p=0
## Wald test = 29003 on 5 df, p=0
## Score (logrank) test = 38248 on 5 df, p=0

summary(coxph(Surv(Time, DSD) ~ offset(mskcc_post.linpred.dresden) + AgeCent + SexM + SizeCent + A2 + A4, data = data.dresden))

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, : Ran out of
iterations and did not converge
## Error in fitter(X, Y, strats, offset, init, control, weights = weights, : NA/NaN/Inf in
foreign function call (arg 6)

summary(coxph(Surv(Time, DSD) ~ offset(gg.linpred.dresden) + AgeCent + SexM + SizeCent + A2 + A4, data = data.dresden))

## Call:
## coxph(formula = Surv(Time, DSD) ~ offset(gg.linpred.dresden) +
##       AgeCent + SexM + SizeCent + A2 + A4, data = data.dresden)
##
## n= 150, number of events= 112
##
##          coef exp(coef) se(coef)      z Pr(>|z|)
## AgeCent   0.01589   1.01601  0.01103  1.44   0.150
## SexMTRUE   0.46624   1.59399  0.19189  2.43   0.015
## SizeCent   0.00808   1.00812  0.00918  0.88   0.378
## A2TRUE    -0.08110   0.92210  0.21938 -0.37   0.712
## A4TRUE     0.08918   1.09328  0.32044  0.28   0.781
##
##          exp(coef) exp(-coef) lower .95 upper .95
## AgeCent     1.016     0.984   0.994   1.04
## SexMTRUE     1.594     0.627   1.094   2.32
## SizeCent     1.008     0.992   0.990   1.03
## A2TRUE       0.922     1.084   0.600   1.42
## A4TRUE       1.093     0.915   0.583   2.05
##
## Concordance= 0.595 (se = 0.031 )
## Rsquare= 0.053 (max possible= 0.998 )
## Likelihood ratio test= 8.1 on 5 df, p=0.151
## Wald test = 8 on 5 df, p=0.156
## Score (logrank) test = 8.1 on 5 df, p=0.151
```

Still strong evidence of misspecification or poor fit. However, the above calibration slope was not significantly different from 1. Hmm. This doesn't necessarily sink the method, but will need checking as we go along.

#### 4.4 Altman method 3 (D)

```
library(Hmisc)
rcorr.cens(gg.linpred.glasgow, Surv(data.glasgow$Time, data.glasgow$DSD))

##          C Index          Dxy          S.D.          n          missing
##    3.928e-01    -2.145e-01    4.883e-02    1.890e+02    0.000e+00
## uncensored Relevant Pairs Concordant Uncertain
##    1.610e+02    2.891e+04    1.135e+04    6.608e+03

rcorr.cens(mskcc_pre.linpred.glasgow, Surv(data.glasgow$Time, data.glasgow$DSD))

##          C Index          Dxy          S.D.          n          missing
##    0.4150      -0.1699      0.0514      189.0000      0.0000
## uncensored Relevant Pairs Concordant Uncertain
##    161.0000    28906.0000    11997.0000    6608.0000

rcorrp.cens(gg.linpred.glasgow, mskcc_pre.linpred.glasgow, Surv(data.glasgow$Time, data.glasgow$DSD))

##          Dxy          S.D. x1 more concordant
##    1.272e-01    5.245e-02    5.636e-01
## x2 more concordant          n          missing
##    4.364e-01    1.890e+02    0.000e+00
## uncensored Relevant Pairs Uncertain
##    1.610e+02    2.891e+04    6.608e+03
##          C X1          C X2          Dxy X1
##    3.928e-01    4.150e-01    -2.145e-01
##          Dxy X2
##    -1.699e-01

rcorr.cens(gg.linpred.apgi, Surv(data.apgi$Time, data.apgi$DSD))

##          C Index          Dxy          S.D.          n          missing
##    0.42092      -0.15815    0.08359      75.00000      0.00000
## uncensored Relevant Pairs Concordant Uncertain
##    51.00000    4464.00000    1879.00000    1086.00000

rcorr.cens(mskcc_pre.linpred.apgi, Surv(data.apgi$Time, data.apgi$DSD))

##          C Index          Dxy          S.D.          n          missing
##    5.251e-01    5.018e-02    8.872e-02    7.500e+01    0.000e+00
## uncensored Relevant Pairs Concordant Uncertain
##    5.100e+01    4.464e+03    2.344e+03    1.086e+03

rcorrp.cens(gg.linpred.apgi, mskcc_pre.linpred.apgi, Surv(data.apgi$Time, data.apgi$DSD))

##          Dxy          S.D. x1 more concordant
##    -0.06855      0.08827      0.46550
## x2 more concordant          n          missing
##    0.53405      75.00000      0.00000
## uncensored Relevant Pairs Uncertain
##    51.00000    4464.00000    1086.00000
```

```
##           C X1           C X2           Dxy X1
##           0.42092         0.52509         -0.15815
##           Dxy X2
##           0.05018

rccorr.cens(gg.linpred.dresden, Surv(data.dresden$Time, data.dresden$DSD))

##           C Index           Dxy           S.D.           n           missing
##           4.553e-01         -8.942e-02         5.994e-02         1.500e+02         0.000e+00
##           uncensored Relevant Pairs Concordant Uncertain
##           1.120e+02         1.745e+04         7.943e+03         4.890e+03

rccorr.cens(mskcc_pre.linpred.dresden, Surv(data.dresden$Time, data.dresden$DSD))

##           C Index           Dxy           S.D.           n           missing
##           4.823e-01         -3.542e-02         5.516e-02         1.500e+02         0.000e+00
##           uncensored Relevant Pairs Concordant Uncertain
##           1.120e+02         1.745e+04         8.414e+03         4.890e+03

rccorrp.cens(gg.linpred.dresden, mskcc_pre.linpred.dresden, Surv(data.dresden$Time, data.dresden$DSD))

##           Dxy           S.D. x1 more concordant
##           1.765e-02         5.548e-02         5.087e-01
## x2 more concordant           n           missing
##           4.910e-01         1.500e+02         0.000e+00
##           uncensored Relevant Pairs Uncertain
##           1.120e+02         1.745e+04         4.890e+03
##           C X1           C X2           Dxy X1
##           4.553e-01         4.823e-01         -8.942e-02
##           Dxy X2
##           -3.542e-02

library(survcomp)
concordance.index(gg.linpred.glasgow, data.glasgow$Time, data.glasgow$DSD, method = "noether")

## $c.index
## [1] 0.6086
##
## $se
## [1] 0.02411
##
## $lower
## [1] 0.5613
##
## $upper
## [1] 0.6558
##
## $p.value
## [1] 6.667e-06
##
## $n
## [1] 189
##
## $data
## $data$x
```

```

##      [1]  0.80162  0.34886  0.09981  0.01110 -0.04435  0.18851 -0.28454
##      [8] -0.03326  0.62420 -0.04435  0.38434  0.51708  0.93468  0.04435
##     [15] -0.41760  0.38434 -0.28454  0.27789  0.42870  0.56144  0.33967
##     [22] -0.04435  0.14416  0.04435  0.54402  0.25096  0.65046  0.14416
##     [29] -0.13306  0.05545  0.01110 -0.08871 -0.03326 -0.09535 -0.15147
##     [36] -0.28454  0.29563  0.42838  0.51708  0.25096  0.09981 -0.28454
##     [43] -0.24018  0.56176  0.80162 -0.32002  0.71291  0.33967  0.01110
##     [50]  0.18851 -0.11532 -0.41760  0.47305  0.13306  0.70404  0.47273
##     [57] -0.37324  0.71291  0.80162  0.04435  0.28676  0.56144  1.06774
##     [64] -0.13306  0.71291  0.84597  0.71291  0.38434  0.18851  0.38434
##     [71]  0.29563 -0.05987  0.47273  0.00000  0.32157 -0.28454  0.36660
##     [78]  0.19516  0.42838  0.42838  0.42870  0.02884 -0.02439  0.60611
##     [85] -0.41760  0.71291  0.03771 -0.42647  0.56176 -0.37324  0.32857
##     [92]  0.73952  0.42838  0.04435 -0.03326  0.01110  0.65014 -0.37324
##     [99]  0.51740  0.38434  0.51740  0.00000 -0.03548  0.04435 -0.01774
##    [106] -0.04435  0.42838  0.47305  0.71291 -0.46195  0.26870 -0.07761
##    [113]  0.38402 -0.41760  0.75726  0.01110 -0.42647 -0.03326  0.71291
##    [120]  0.47273 -0.20470  0.47273  0.47305  0.00000  0.65969 -0.03326
##    [127]  0.33999 -0.12196 -0.13306  0.38402  0.31306 -0.42647  0.29563
##    [134]  0.27789 -0.05987  0.66856  0.54370  0.51708  0.01110  0.55034
##    [141]  0.38402  0.36660 -0.12196  0.80162 -0.19583  0.33112 -0.03326
##    [148]  0.89033  0.80162  0.14416 -0.04435  0.01110 -0.12196  0.84597
##    [155]  0.35518  0.33999  0.22145  0.62420 -0.37324  0.60579 -0.32889
##    [162]  0.18851  0.14416  0.75726 -0.37324 -0.21067  0.60579  0.29563
##    [169]  0.27722  0.33967  0.78388  0.42838  0.84597 -0.08871  0.52595
##    [176] -0.31115  0.38434  0.05545  0.33999  0.66856  0.05545  0.18851
##    [183] -0.32889 -0.28454 -0.07761 -0.04435  0.33999  0.38434  0.01997
##
## $data$surv.time
##      [1] 233.456 266.937  40.482 216.106 2611.537 410.906 473.912
##      [8] 610.881 1380.949 1443.651 959.694 147.013  10.044 915.256
##     [15] 161.319 2884.562 136.056 3530.750 450.475 143.969 17.349
##     [22] 1104.881 267.850 655.319  38.656 283.069 1018.743 812.681
##     [29] 1452.782 1420.518 1342.294 900.950 308.332 336.943 1116.143
##     [36] 1016.613 423.994 741.762 1921.519 597.488  8.218 25.263
##     [43] 803.550 350.031 394.774 299.201 148.231 1094.837 25.263
##     [50] 312.593 436.169 751.806 492.174  11.262 416.994 809.637
##     [57] 806.594 142.143 621.838 316.550 2749.419 136.056 49.613
##     [64] 379.556 210.932 256.588 253.544 705.237 1254.938 510.437
##     [71] 890.906 116.576 178.668 519.568  14.306 702.193 148.231
##     [78] 560.050 113.532  9.131 152.188 1102.751 375.294 343.944
##     [85] 353.075 2830.688 185.669 1310.943 1491.438 576.182 663.538
##     [92] 408.776 548.788 781.331 467.824 636.144  76.094 468.738
##     [99]  9.131 501.306 248.674 837.944 2221.938 2556.750 2587.188
##    [106] 286.113 542.701 534.787 396.601 856.207 139.099 29.524
##    [113] 629.143 1205.325 868.382 395.688 1046.137  43.526 862.294
##    [120] 235.282 177.451 118.706 553.049 903.081 563.094 404.819
##    [127] 565.224 203.018 190.843 398.731 413.037 786.201 497.044
##    [134] 942.649 883.601 257.806 344.857 191.756  45.656 730.500
##    [141] 103.487 1004.438 1461.000 974.000 118.706 1156.625 703.106
##    [148] 233.151 581.356 1552.312 1156.625 1795.812 2343.688 175.624
##    [155] 1649.712 1978.438 487.000 602.663 1552.312  24.350 608.750
##    [162] 319.594  76.094 730.500 1217.500 1461.000  91.312  30.438

```



```

## [169] 182.625 517.438 213.062 760.938 243.500 943.562 152.188
## [176] 1126.188 760.938 1068.356 1129.231 243.500 273.938 362.206
## [183] 1007.481 669.625 1187.062 273.938 608.750 1065.312 426.125
##
## $data$surv.event
## [1] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [12] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [23] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [34] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE
## [45] TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE FALSE TRUE
## [56] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [67] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [78] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [89] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [100] TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [111] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [122] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [133] TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [144] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [155] FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE
## [166] FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [177] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE
## [188] FALSE TRUE
##
##
## $comppairs
## [1] 28512

concordance.index(mskcc_pre.linpred.glasgow, data.glasgow$Time, data.glasgow$DSD, method = "noether")

## $c.index
## [1] 0.5851
##
## $se
## [1] 0.0251
##
## $lower
## [1] 0.5359
##
## $upper
## [1] 0.6343
##
## $p.value
## [1] 0.0006962
##
## $n
## [1] 189
##
## $data
## $data$x
## [1] 158.0 160.9 161.6 161.9 158.5 162.1 164.5 164.5 161.4 146.8 159.6
## [12] 161.4 161.0 149.4 164.7 158.2 161.7 159.0 157.7 162.4 160.8 158.5
## [23] 162.6 159.0 159.1 164.5 146.2 160.7 152.3 160.5 149.3 158.7 161.3
## [34] 187.6 162.4 163.1 158.1 151.0 161.3 165.4 160.2 150.4 162.0 161.1

```

```

## [45] 159.0 162.9 159.4 161.3 161.6 162.4 158.6 162.4 161.2 159.1 160.5
## [56] 160.6 162.1 158.5 160.9 160.4 147.4 162.1 159.1 165.0 159.7 158.4
## [67] 160.3 158.1 162.0 158.0 161.2 162.6 163.6 161.4 161.5 163.2 158.5
## [78] 149.9 160.7 161.3 161.0 154.0 161.3 162.4 162.8 158.5 161.3 160.7
## [89] 157.5 161.7 158.7 162.1 161.1 157.4 162.2 161.5 161.0 162.8 159.9
## [100] 159.7 160.1 158.2 157.7 158.3 159.5 161.6 161.3 158.5 159.7 162.0
## [111] 162.4 162.8 162.8 161.0 159.6 162.7 161.5 161.0 148.1 161.8 161.3
## [122] 165.6 159.7 158.8 159.2 160.8 161.3 164.6 165.1 163.6 161.8 162.2
## [133] 159.5 161.3 163.0 161.4 161.9 160.9 160.4 159.4 161.7 159.6 166.0
## [144] 157.8 161.9 159.4 163.0 159.8 160.1 160.2 161.3 161.1 166.1 158.7
## [155] 160.3 149.3 162.2 163.8 164.3 162.6 161.9 159.1 163.9 147.1 161.5
## [166] 201.9 162.7 157.7 161.5 161.4 158.7 162.9 158.0 162.0 163.3 163.5
## [177] 158.1 164.6 158.5 159.6 161.7 164.7 162.1 162.3 162.6 160.0 158.7
## [188] 158.5 160.7
##
## $data$surv.time
## [1] 233.456 266.937 40.482 216.106 2611.537 410.906 473.912
## [8] 610.881 1380.949 1443.651 959.694 147.013 10.044 915.256
## [15] 161.319 2884.562 136.056 3530.750 450.475 143.969 17.349
## [22] 1104.881 267.850 655.319 38.656 283.069 1018.743 812.681
## [29] 1452.782 1420.518 1342.294 900.950 308.332 336.943 1116.143
## [36] 1016.613 423.994 741.762 1921.519 597.488 8.218 25.263
## [43] 803.550 350.031 394.774 299.201 148.231 1094.837 25.263
## [50] 312.593 436.169 751.806 492.174 11.262 416.994 809.637
## [57] 806.594 142.143 621.838 316.550 2749.419 136.056 49.613
## [64] 379.556 210.932 256.588 253.544 705.237 1254.938 510.437
## [71] 890.906 116.576 178.668 519.568 14.306 702.193 148.231
## [78] 560.050 113.532 9.131 152.188 1102.751 375.294 343.944
## [85] 353.075 2830.688 185.669 1310.943 1491.438 576.182 663.538
## [92] 408.776 548.788 781.331 467.824 636.144 76.094 468.738
## [99] 9.131 501.306 248.674 837.944 2221.938 2556.750 2587.188
## [106] 286.113 542.701 534.787 396.601 856.207 139.099 29.524
## [113] 629.143 1205.325 868.382 395.688 1046.137 43.526 862.294
## [120] 235.282 177.451 118.706 553.049 903.081 563.094 404.819
## [127] 565.224 203.018 190.843 398.731 413.037 786.201 497.044
## [134] 942.649 883.601 257.806 344.857 191.756 45.656 730.500
## [141] 103.487 1004.438 1461.000 974.000 118.706 1156.625 703.106
## [148] 233.151 581.356 1552.312 1156.625 1795.812 2343.688 175.624
## [155] 1649.712 1978.438 487.000 602.663 1552.312 24.350 608.750
## [162] 319.594 76.094 730.500 1217.500 1461.000 91.312 30.438
## [169] 182.625 517.438 213.062 760.938 243.500 943.562 152.188
## [176] 1126.188 760.938 1068.356 1129.231 243.500 273.938 362.206
## [183] 1007.481 669.625 1187.062 273.938 608.750 1065.312 426.125
##
## $data$surv.event
## [1] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [12] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [23] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [34] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE
## [45] TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE FALSE TRUE
## [56] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [67] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [78] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

```

```

## [89] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE
## [100] TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [111] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [122] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [133] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [144] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [155] FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE FALSE
## [166] FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [177] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE
## [188] FALSE TRUE
##
##
## $comppairs
## [1] 28884

concordance.index(gg.linpred.apgi, data.apgi$Time, data.apgi$DSD, method = "noether")

## $c.index
## [1] 0.5798
##
## $se
## [1] 0.03978
##
## $lower
## [1] 0.5019
##
## $upper
## [1] 0.6578
##
## $p.value
## [1] 0.04476
##
## $n
## [1] 75
##
## $data
## $data$x
## [1] 0.188513 0.384341 0.428695 0.081914 0.115320 0.046581 0.505986
## [8] 0.757265 0.543345 0.073193 0.384341 0.314781 0.775007 0.295633
## [15] 0.055451 -0.195828 0.099805 0.384341 0.314781 0.295633 0.757265
## [22] 0.046581 0.106449 0.357729 -0.382114 0.159674 0.608188 0.046581
## [29] 0.011098 0.437246 0.011098 0.099805 0.643351 -0.006644 0.133062
## [36] -0.122785 -0.044354 -0.444210 -0.390985 -0.104223 0.177416 0.046581
## [43] 0.126418 0.561437 0.144159 0.295633 0.099805 0.517403 -0.461951
## [50] -0.077610 0.739172 -0.044354 0.232867 0.569807 0.758320 -0.461951
## [57] 0.428695 0.339987 -0.328890 0.000000 -0.033256 -0.328890 0.492197
## [64] 0.845973 0.801619 0.517403 0.170772 -0.088708 -0.070966 -0.113914
## [71] -0.382114 0.339317 0.676608 0.366280 -0.044354
##
## $data$surv.time
## [1] 525 2848 1779 1279 37 436 1048 476 135 315 452 346 858 549
## [15] 1235 715 949 269 410 516 359 750 1081 835 768 989 71 1574
## [29] 209 313 312 1504 553 718 1588 237 1268 518 1533 1430 610 1461
## [43] 106 325 142 1171 365 281 1278 611 976 1407 427 1094 655 913

```

```

## [57] 796 480 162 1176 852 521 482 695 412 1030 305 869 892 241
## [71] 911 120 950 545 709
##
## $data$surv.event
## [1] 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 0
## [36] 1 0 1 0 0 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 0 0 1
## [71] 0 1 0 1 1
##
##
## $comppairs
## [1] 4422

concordance.index(mskcc_pre.linpred.apgi, data.apgi$Time, data.apgi$DSD, method = "noether")

## $c.index
## [1] 0.4745
##
## $se
## [1] 0.04286
##
## $lower
## [1] 0.3904
##
## $upper
## [1] 0.5585
##
## $p.value
## [1] 0.5511
##
## $n
## [1] 75
##
## $data
## $data$x
## [1] 159.7 157.3 159.9 240.6 157.7 160.3 159.6 157.5 156.8 158.9 145.4
## [12] 210.1 157.1 159.2 160.0 159.6 161.1 158.4 210.1 160.3 157.1 158.5
## [23] 159.5 158.8 160.8 158.5 208.2 148.1 160.0 158.7 157.3 157.7 209.5
## [34] 160.7 158.0 209.0 158.9 156.5 157.2 157.5 157.9 158.1 157.2 159.3
## [45] 157.2 159.5 159.1 159.7 158.5 161.3 149.1 158.3 158.8 208.2 210.6
## [56] 158.3 158.3 158.9 157.1 159.3 159.1 160.3 209.1 157.5 156.5 159.3
## [67] 158.8 160.7 160.7 209.0 159.3 158.7 209.0 157.3 158.0
##
## $data$surv.time
## [1] 525 2848 1779 1279 37 436 1048 476 135 315 452 346 858 549
## [15] 1235 715 949 269 410 516 359 750 1081 835 768 989 71 1574
## [29] 209 313 312 1504 553 718 1588 237 1268 518 1533 1430 610 1461
## [43] 106 325 142 1171 365 281 1278 611 976 1407 427 1094 655 913
## [57] 796 480 162 1176 852 521 482 695 412 1030 305 869 892 241
## [71] 911 120 950 545 709
##
## $data$surv.event
## [1] 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 0
## [36] 1 0 1 0 0 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 0 0 1
## [71] 0 1 0 1 1

```

```
##
##
## $comppairs
## [1] 4384

concordance.index(gg.linpred.dresden, data.dresden$Time, data.dresden$DSD, method = "noether")

## $c.index
## [1] 0.5457
##
## $se
## [1] 0.0297
##
## $lower
## [1] 0.4874
##
## $upper
## [1] 0.6039
##
## $p.value
## [1] 0.1243
##
## $n
## [1] 150
##
## $data
## $data$x
## [1] -0.088708 0.099805 -0.461951 0.517083 0.473049 0.428695 0.499342
## [8] 0.011098 -0.328890 0.845973 0.295633 0.295314 0.384341 0.099805
## [15] 0.011098 0.580905 0.295633 0.286763 0.339987 -0.033256 0.705447
## [22] 0.000000 0.561437 -0.417598 0.303364 0.384341 0.428695 -0.033256
## [29] 0.525134 0.144159 -0.033256 0.845973 0.490791 0.339987 0.384341
## [36] 0.472729 0.845973 -0.133062 -0.088708 0.055451 0.055451 0.554963
## [43] 0.677428 0.428695 0.099805 -0.077610 -0.077610 -0.050998 0.801619
## [50] -0.033256 0.517403 0.863714 0.250960 0.321575 0.055451 0.384341
## [57] 0.428375 0.845973 0.668557 -0.059869 0.082064 0.055451 0.384341
## [64] 0.650816 0.144159 0.188513 0.314781 0.517403 0.144159 0.011098
## [71] 0.428695 0.384021 0.845973 -0.121964 0.011098 0.561757 0.295633
## [78] -0.033256 0.019968 1.087210 -0.006644 0.339667 0.819360 0.561437
## [85] 0.055451 0.295314 0.517403 0.384341 0.401763 0.428375 0.428695
## [92] -0.033256 1.105303 0.934681 0.490471 0.055451 0.339987 0.419504
## [99] 0.073193 -0.033256 0.339987 0.357729 -0.346631 -0.033256 0.472729
## [106] 0.002227 -0.097579 0.011098 -0.328890 0.402083 0.195157 -0.033256
## [113] 0.499661 0.473049 0.428375 0.473049 0.446437 0.099805 0.561437
## [120] -0.006644 0.776733 0.000000 0.339987 0.473049 0.794474 0.159674
## [127] -0.006644 0.011098 0.055451 0.775007 -0.006644 0.473049 0.339987
## [134] 0.437566 0.392892 0.011098 0.384341 0.019968 -0.077610 0.055451
## [141] 0.650145 0.313375 0.561757 0.011098 -0.035483 0.295314 -0.035483
## [148] 0.099805 0.473049 0.019148
##
## $data$surv.time
## [1] 475 319 478 292 266 4190 511 1211 1379 844 737 338 583 1379
## [15] 3691 496 360 1173 379 391 522 1486 27 891 356 123 2906 500
## [29] 1450 747 357 10 2017 1256 1517 152 1061 317 698 435 987 975
```

```

## [43] 604 859 427 1632 4059 521 526 471 309 305 559 110 844 389
## [57] 349 1098 2353 169 420 1292 648 1022 219 392 540 717 319 1461
## [71] 643 293 283 55 781 322 605 981 1920 375 704 886 735 13
## [85] 1823 178 802 1096 453 69 288 707 522 254 727 153 652 1177
## [99] 348 445 442 239 218 419 392 524 154 614 1727 1422 478 516
## [113] 184 330 303 502 1383 491 100 1101 138 17 538 278 761 1497
## [127] 49 698 641 71 1109 597 1204 304 1030 873 467 188 18 338
## [141] 279 308 448 268 923 1087 376 1084 252 745
##
## $data$surv.event
## [1] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [12] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [23] FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE
## [34] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [45] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [56] TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [67] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [78] FALSE FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE FALSE
## [89] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE
## [100] TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE FALSE
## [111] FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [122] FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE
## [133] FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [144] TRUE TRUE FALSE FALSE TRUE TRUE FALSE
##
##
## $comppairs
## [1] 17106

concordance.index(mskcc_pre.linpred.dresden, data.dresden$Time, data.dresden$DSD, method = "noether")

## $c.index
## [1] 0.518
##
## $se
## [1] 0.02694
##
## $lower
## [1] 0.4652
##
## $upper
## [1] 0.5708
##
## $p.value
## [1] 0.5043
##
## $n
## [1] 150
##
## $data
## $data$x
## [1] 147.5 152.0 150.8 188.4 148.2 147.3 153.6 154.3 150.3 148.5 146.6
## [12] 150.5 147.9 149.7 151.7 203.0 146.7 147.9 146.3 149.1 201.5 147.8
## [23] 149.6 153.5 202.2 150.9 148.7 150.9 239.0 153.5 150.4 147.9 149.0

```

```

## [34] 146.5 149.3 149.5 146.9 188.0 148.9 152.9 149.3 198.5 147.5 147.7
## [45] 152.3 150.7 152.1 187.5 147.9 150.3 148.2 147.7 154.9 150.1 150.1
## [56] 147.9 150.3 149.3 149.3 151.1 150.7 150.7 149.4 148.9 149.3 152.8
## [67] 201.7 151.1 150.1 151.9 148.1 152.4 147.5 153.3 152.9 147.4 147.5
## [78] 150.3 150.3 199.9 153.3 185.3 146.9 150.9 151.5 151.5 146.7 147.1
## [89] 150.5 150.0 147.0 149.1 200.7 182.9 151.2 150.3 146.7 150.3 151.7
## [100] 151.7 146.2 146.3 151.3 150.0 149.7 150.3 146.5 150.4 150.9 147.3
## [111] 182.7 153.1 148.9 147.5 150.1 146.3 146.7 150.3 149.7 149.2 197.6
## [122] 148.2 149.9 146.7 201.2 147.4 148.4 149.5 150.5 145.9 154.3 146.7
## [133] 147.5 146.7 151.2 149.5 147.3 149.9 149.2 150.8 152.1 146.9 147.3
## [144] 149.7 146.3 151.3 147.3 152.4 185.3 237.7
##
## $data$surv.time
## [1] 475 319 478 292 266 4190 511 1211 1379 844 737 338 583 1379
## [15] 3691 496 360 1173 379 391 522 1486 27 891 356 123 2906 500
## [29] 1450 747 357 10 2017 1256 1517 152 1061 317 698 435 987 975
## [43] 604 859 427 1632 4059 521 526 471 309 305 559 110 844 389
## [57] 349 1098 2353 169 420 1292 648 1022 219 392 540 717 319 1461
## [71] 643 293 283 55 781 322 605 981 1920 375 704 886 735 13
## [85] 1823 178 802 1096 453 69 288 707 522 254 727 153 652 1177
## [99] 348 445 442 239 218 419 392 524 154 614 1727 1422 478 516
## [113] 184 330 303 502 1383 491 100 1101 138 17 538 278 761 1497
## [127] 49 698 641 71 1109 597 1204 304 1030 873 467 188 18 338
## [141] 279 308 448 268 923 1087 376 1084 252 745
##
## $data$surv.event
## [1] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [12] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [23] FALSE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE FALSE TRUE
## [34] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [45] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [56] TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [67] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [78] FALSE FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE FALSE
## [89] TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE
## [100] TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE FALSE
## [111] FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [122] FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE
## [133] FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE
## [144] TRUE TRUE FALSE FALSE TRUE TRUE FALSE
##
##
## $comppairs
## [1] 17236

cindex.comp(concordance.index(gg.linpred.glasgow, data.glasgow$Time, data.glasgow$DSD, method = "noether

## $p.value
## [1] 0.283
##
## $cindex1
## [1] 0.6086
##
## $cindex2
## [1] 0.5851

```

```

cindex.comp(concordance.index(gg.linpred.apgi, data.apgi$Time, data.apgi$DSD, method = "noether"), concordance.index(gg.linpred.dresden, data.dresden$Time, data.dresden$DSD, method = "noether"))

## $p.value
## [1] 0.04175
##
## $cindex1
## [1] 0.5798
##
## $cindex2
## [1] 0.4745

cindex.comp(concordance.index(gg.linpred.dresden, data.dresden$Time, data.dresden$DSD, method = "noether"), concordance.index(gg.linpred.apgi, data.apgi$Time, data.apgi$DSD, method = "noether"))

## $p.value
## [1] 0.2626
##
## $cindex1
## [1] 0.5457
##
## $cindex2
## [1] 0.518

```

## 4.5 Altman method 4 (D,C)

```

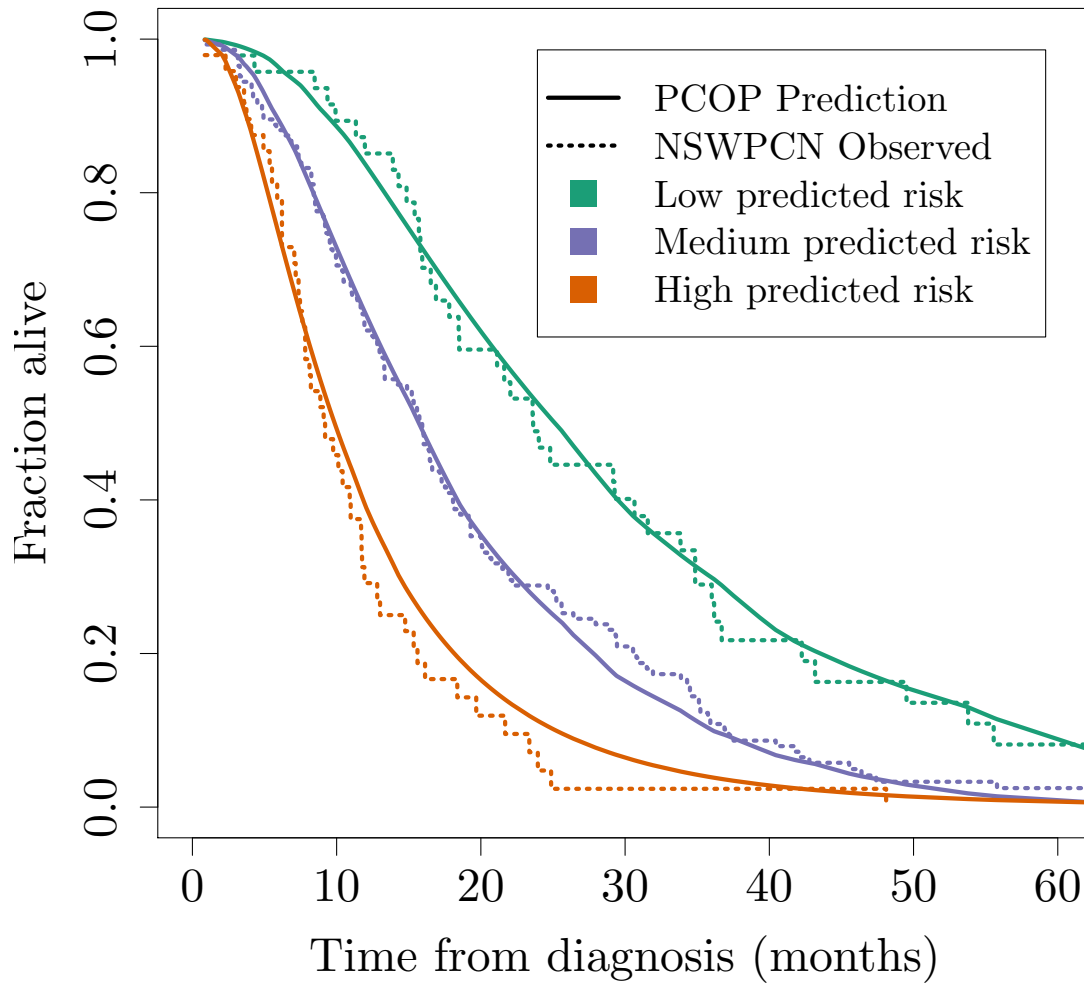
group_quantiles = c(0, 0.2, 0.8, 1)
gg.groups.nswpcn = cut(gg.linpred.nswpcn, quantile(gg.linpred.nswpcn, group_quantiles), labels = FALSE)
temp.alpha = 0.1

temp.km = survfit(Surv(data.nswpcn$Time, data.nswpcn$DSD) ~ gg.groups.nswpcn, conf.int = 1-temp.alpha)
temp.km = data.frame(surv = temp.km$surv, group = rep(gsub(".*=", "", names(temp.km$strata)), temp.km$strata))
temp.pred = summary(fit.gg, newdata = data.nswpcn, ci = FALSE)
temp.pred.times = temp.pred[[1]][,1]
temp.pred.ests = sapply(temp.pred, function(x) x[,2])
temp.pred.ests = tapply(1:ncol(temp.pred.ests), gg.groups.nswpcn, function(is) apply(temp.pred.ests[,is], 1, function(x) x[is]))
temp.pred.lower = sapply(temp.pred.ests, function(x) x[1,])
temp.pred.meds = sapply(temp.pred.ests, function(x) x[2,])
temp.pred.upper = sapply(temp.pred.ests, function(x) x[3,])
temp.pred = data.frame(surv = as.vector(temp.pred.meds), group = rep(colnames(temp.pred.meds), each = nrow(temp.pred.meds)))
temp.data = rbind(temp.km, temp.pred)
temp.data$time = temp.data$time / 365.25 * 12

plot(0 ~ 0, type = "n", xlim = c(0, 5*12), ylim = c(0, 1), xlab = "Time from diagnosis (months)", ylab = "Survival probability")
temp.pal = brewer.pal(length(unique(gg.groups.nswpcn)), "Dark2")[c(1, 3, 2)]
names(temp.pal) = sort(unique(gg.groups.nswpcn))
for (temp.i in factor(sort(unique(gg.groups.nswpcn))))
{
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$time < 5*12,], col = temp.pal[1])
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$time >= 5*12,], col = temp.pal[2])
}
legend("topright", inset = 0.05, legend = c("PCOP Prediction", "NSWPCN Observed", "Low predicted risk", "High predicted risk"), col = temp.pal, bty = "n")

```





```
summary(coxph(Surv(data.nswpcn$Time, data.nswpcn$DSD) ~ factor(gg.groups.nswpcn)))
```

## Call:  
## coxph(formula = Surv(data.nswpcn\$Time, data.nswpcn\$DSD) ~ factor(gg.groups.nswpcn))  
##  
## n= 239, number of events= 230  
## (1 observation deleted due to missingness)  
##  
##

	coef	exp(coef)	se(coef)	z	Pr(> z )
## factor(gg.groups.nswpcn)2	0.532	1.703	0.176	3.03	0.0025
## factor(gg.groups.nswpcn)3	1.328	3.775	0.219	6.06	1.3e-09

##

	exp(coef)	exp(-coef)	lower .95	upper .95
## factor(gg.groups.nswpcn)2	1.70	0.587	1.21	2.4
## factor(gg.groups.nswpcn)3	3.78	0.265	2.46	5.8

##

```
## Concordance= 0.618 (se = 0.019 )
## Rsquare= 0.138 (max possible= 1 )
## Likelihood ratio test= 35.5 on 2 df, p=1.96e-08
## Wald test = 37.9 on 2 df, p=6.01e-09
## Score (logrank) test = 40.7 on 2 df, p=1.46e-09
```

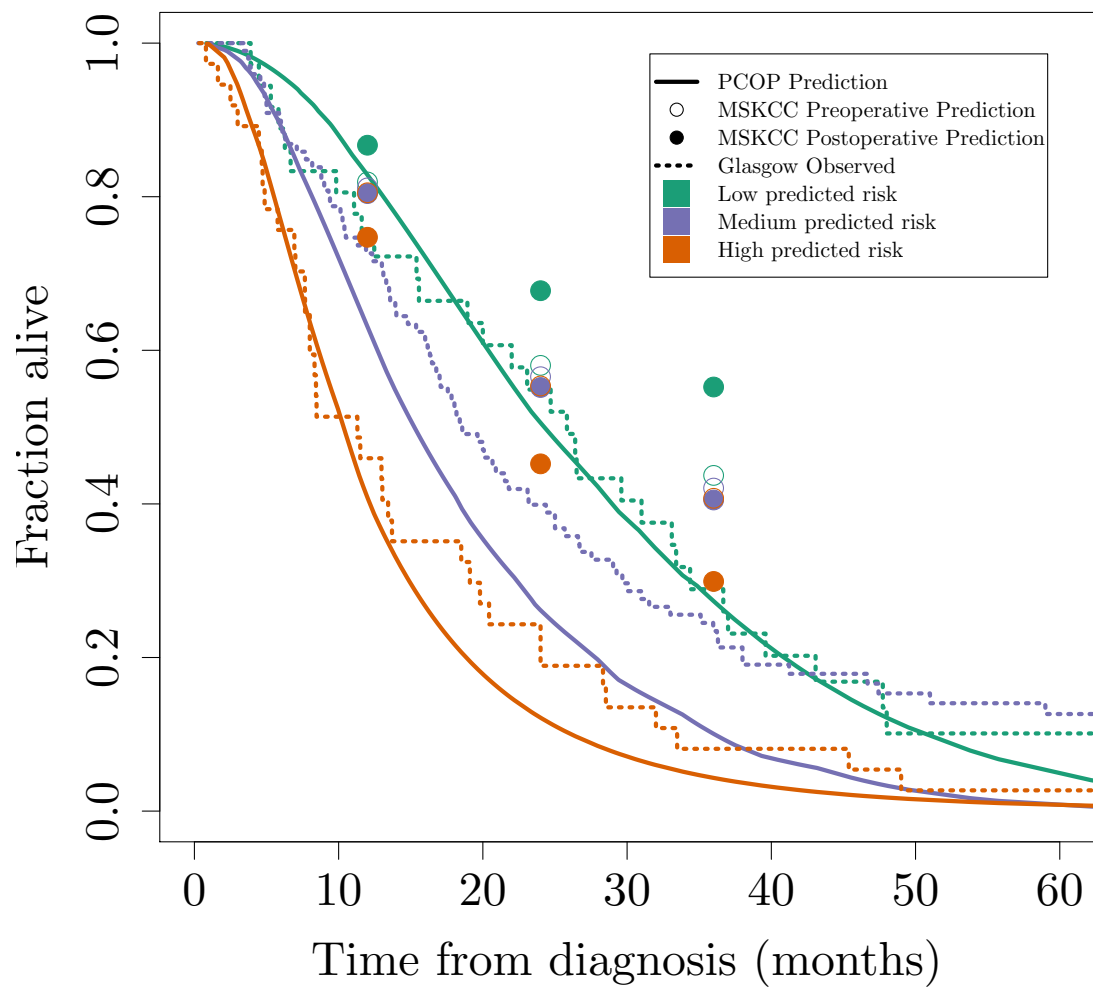
```

mskcc_pre.groups.glasgow = cut(mskcc_pre.linpred.glasgow, quantile(mskcc_pre.linpred.glasgow, group_quantiles), labels = FALSE)
mskcc_post.groups.glasgow = cut(mskcc_post.linpred.glasgow, quantile(mskcc_post.linpred.glasgow, group_quantiles), labels = FALSE)
gg.groups.glasgow = cut(gg.linpred.glasgow, quantile(gg.linpred.glasgow, group_quantiles), labels = FALSE)

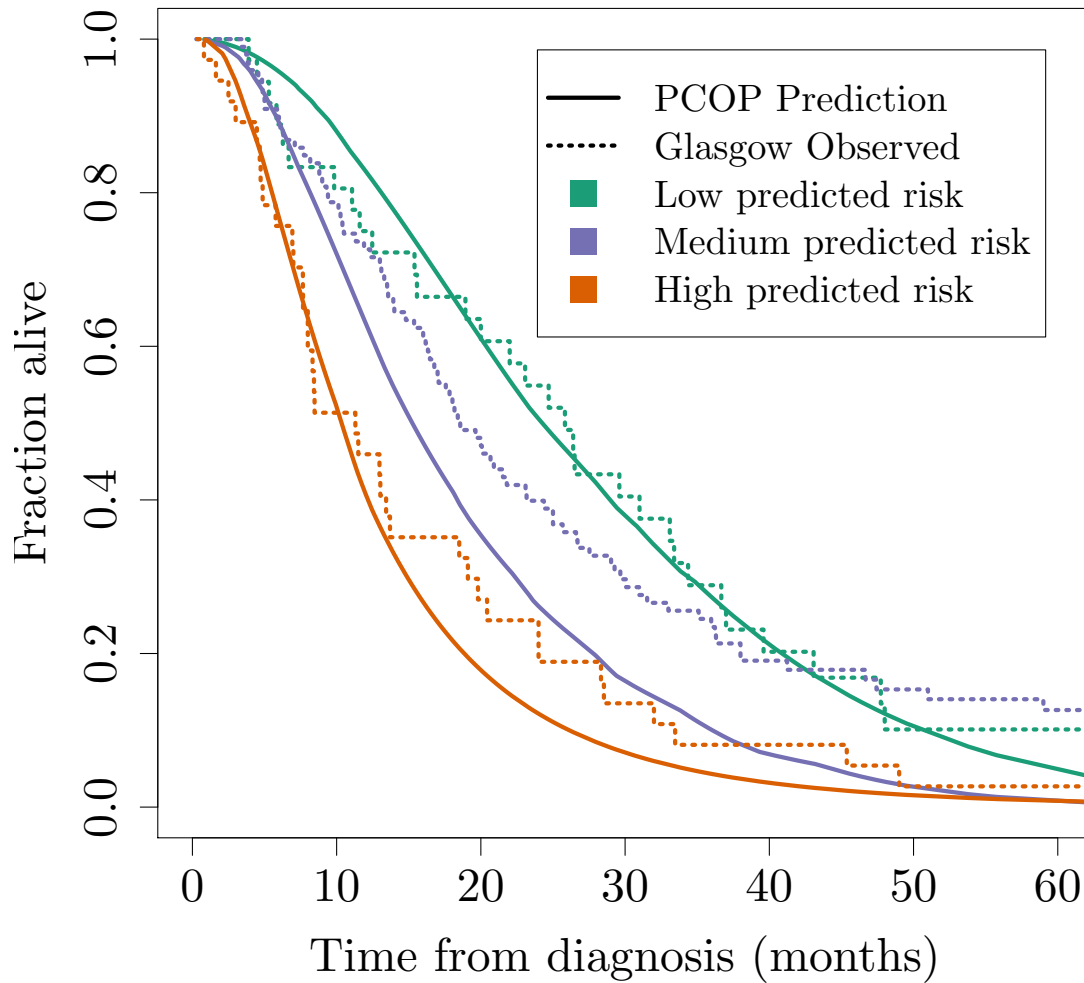
temp.km = survfit(Surv(data.glasgow$Time, data.glasgow$DSD) ~ gg.groups.glasgow, conf.int = 1-temp.alpha)
temp.km = data.frame(surv = temp.km$surv, group = rep(gsub(".*=", "", names(temp.km$strata)), temp.km$strata))
temp.pred = summary(fit.gg, newdata = data.glasgow, ci = FALSE)
temp.pred.times = temp.pred[[1]][,1]
temp.pred.ests = sapply(temp.pred, function(x) x[,2])
temp.pred.ests = tapply(1:ncol(temp.pred.ests), gg.groups.glasgow, function(is) apply(temp.pred.ests[,is], 1, FUN = function(x) x[is,]))
temp.pred.lower = sapply(temp.pred.ests, function(x) x[1,])
temp.pred.meds = sapply(temp.pred.ests, function(x) x[2,])
temp.pred.upper = sapply(temp.pred.ests, function(x) x[3,])
temp.pred = data.frame(surv = as.vector(temp.pred.meds), group = rep(colnames(temp.pred.meds), each = nrow(temp.pred.meds)))
temp.data = rbind(temp.km, temp.pred)
temp.data$time = temp.data$time / 365.25 * 12
temp.predpre.12mo = simplify2array(tapply(mskcc_pre.12mo.glasgow, mskcc_pre.groups.glasgow, quantile, probs = temp.pred.times[1:12]))
temp.predpre.24mo = simplify2array(tapply(mskcc_pre.24mo.glasgow, mskcc_pre.groups.glasgow, quantile, probs = temp.pred.times[13:24]))
temp.predpre.36mo = simplify2array(tapply(mskcc_pre.36mo.glasgow, mskcc_pre.groups.glasgow, quantile, probs = temp.pred.times[25:36]))
temp.predpost.12mo = simplify2array(tapply(mskcc_post.12mo.glasgow, mskcc_post.groups.glasgow, quantile, probs = temp.pred.times[37:48]))
temp.predpost.24mo = simplify2array(tapply(mskcc_post.24mo.glasgow, mskcc_post.groups.glasgow, quantile, probs = temp.pred.times[49:60]))
temp.predpost.36mo = simplify2array(tapply(mskcc_post.36mo.glasgow, mskcc_post.groups.glasgow, quantile, probs = temp.pred.times[61:72]))
temp.data2 = data.frame(
  surv = c(temp.predpre.12mo[2,], temp.predpre.24mo[2,], temp.predpre.36mo[2,], temp.predpost.12mo[2,], temp.predpost.24mo[2,], temp.predpost.36mo[2,]),
  group = factor(rep(sort(unique(mskcc_pre.groups.glasgow)), 6)),
  time = rep(c(12, 24, 36), each = 3),
  upper = c(temp.predpre.12mo[3,], temp.predpre.24mo[3,], temp.predpre.36mo[3,], temp.predpost.12mo[3,], temp.predpost.24mo[3,], temp.predpost.36mo[3,]),
  lower = c(temp.predpre.12mo[1,], temp.predpre.24mo[1,], temp.predpre.36mo[1,], temp.predpost.12mo[1,], temp.predpost.24mo[1,], temp.predpost.36mo[1,]),
  est = rep(c("MSKCC Preoperative", "MSKCC Postoperative"), each = 9))

plot(0 ~ 0, type = "n", xlim = c(0, 5*12), ylim = c(0, 1), xlab = "Time from diagnosis (months)", ylab = "Survival Probability")
temp.pal = brewer.pal(length(unique(gg.groups.glasgow)), "Dark2")[c(1, 3, 2)]
names(temp.pal) = sort(unique(gg.groups.glasgow))
for (temp.i in factor(sort(unique(gg.groups.glasgow))))
{
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$time < 12,], col = temp.pal[as.character(temp.i)], lty = 1)
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$time >= 12,], col = temp.pal[as.character(temp.i)], lty = 2)
  points(surv ~ time, temp.data2[as.character(temp.data2$group) == as.character(temp.i) & temp.data2$time < 12,], col = temp.pal[as.character(temp.i)], lty = 1)
  points(surv ~ time, temp.data2[as.character(temp.data2$group) == as.character(temp.i) & temp.data2$time >= 12,], col = temp.pal[as.character(temp.i)], lty = 2)
}
legend("topright", inset = 0.05, legend = c("PCOP Prediction", "MSKCC Preoperative Prediction", "MSKCC Postoperative Prediction"), bty = "n")

```



```
plot(0 ~ 0, type = "n", xlim = c(0, 5*12), ylim = c(0, 1), xlab = "Time from diagnosis (months)", ylab = "Fraction alive")
temp.pal = brewer.pal(length(unique(gg.groups.glasgow)), "Dark2")[c(1, 3, 2)]
names(temp.pal) = sort(unique(gg.groups.glasgow))
for (temp.i in factor(sort(unique(gg.groups.glasgow))))
{
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$group == "Low predicted risk",])
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$group == "Medium predicted risk",])
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$group == "High predicted risk",])
}
legend("topright", inset = 0.05, legend = c("PCOP Prediction", "Glasgow Observed", "Low predicted risk", "Medium predicted risk", "High predicted risk"))
```



```
summary(coxph(Surv(data.glasgow$Time, data.glasgow$DSD) ~ factor(gg.groups.glasgow)))
```

## Call:  
## coxph(formula = Surv(data.glasgow\$Time, data.glasgow\$DSD) ~ factor(gg.groups.glasgow))  
##  
## n= 188, number of events= 160  
## (1 observation deleted due to missingness)  
##  
##

	coef	exp(coef)	se(coef)	z	Pr(> z )
## factor(gg.groups.glasgow)2	0.0794	1.0826	0.2074	0.38	0.7019
## factor(gg.groups.glasgow)3	0.6662	1.9468	0.2438	2.73	0.0063

##

	exp(coef)	exp(-coef)	lower .95	upper .95
## factor(gg.groups.glasgow)2	1.08	0.924	0.721	1.63
## factor(gg.groups.glasgow)3	1.95	0.514	1.207	3.14

##

## Concordance= 0.577 (se = 0.023 )  
## Rsquare= 0.049 (max possible= 0.999 )  
## Likelihood ratio test= 9.37 on 2 df, p=0.00923  
## Wald test = 10.4 on 2 df, p=0.00543  
## Score (logrank) test = 10.8 on 2 df, p=0.00463

```
summary(coxph(Surv(data.glasgow$Time, data.glasgow$DSD) ~ factor(mskcc_pre.groups.glasgow)))
```

```
## Call:
## coxph(formula = Surv(data.glasgow$Time, data.glasgow$DSD) ~ factor(mskcc_pre.groups.glasgow))
##
## n= 188, number of events= 160
## (1 observation deleted due to missingness)
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## factor(mskcc_pre.groups.glasgow)2 0.764      2.147      0.217 3.52  0.00043
## factor(mskcc_pre.groups.glasgow)3 0.762      2.143      0.260 2.93  0.00338
##
##               exp(coef) exp(-coef) lower .95 upper .95
## factor(mskcc_pre.groups.glasgow)2      2.15      0.466      1.40      3.28
## factor(mskcc_pre.groups.glasgow)3      2.14      0.467      1.29      3.57
##
## Concordance= 0.563 (se = 0.023 )
## Rsquare= 0.077 (max possible= 0.999 )
## Likelihood ratio test= 15.1 on 2 df, p=0.000535
## Wald test              = 13.1 on 2 df, p=0.00144
## Score (logrank) test = 13.6 on 2 df, p=0.00109

summary(coxph(Surv(data.glasgow$Time, data.glasgow$DSD) ~ factor(mskcc_post.groups.glasgow)))

## Call:
## coxph(formula = Surv(data.glasgow$Time, data.glasgow$DSD) ~ factor(mskcc_post.groups.glasgow))
##
## n= 188, number of events= 160
## (1 observation deleted due to missingness)
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## factor(mskcc_post.groups.glasgow)2 0.631      1.879      0.218 2.9  0.00378
## factor(mskcc_post.groups.glasgow)3 0.990      2.691      0.261 3.8  0.00015
##
##               exp(coef) exp(-coef) lower .95
## factor(mskcc_post.groups.glasgow)2      1.88      0.532      1.23
## factor(mskcc_post.groups.glasgow)3      2.69      0.372      1.61
##
##               upper .95
## factor(mskcc_post.groups.glasgow)2      2.88
## factor(mskcc_post.groups.glasgow)3      4.49
##
## Concordance= 0.579 (se = 0.023 )
## Rsquare= 0.081 (max possible= 0.999 )
## Likelihood ratio test= 15.8 on 2 df, p=0.000372
## Wald test              = 14.7 on 2 df, p=0.00066
## Score (logrank) test = 15.3 on 2 df, p=0.000484
```

```
mskcc_pre.groups.apgi = cut(mskcc_pre.linpred.apgi, quantile(mskcc_pre.linpred.apgi, group_quantiles), 1)
mskcc_post.groups.apgi = cut(mskcc_post.linpred.apgi, quantile(mskcc_post.linpred.apgi, group_quantiles), 1)
gg.groups.apgi = cut(gg.linpred.apgi, quantile(gg.linpred.apgi, group_quantiles), labels = FALSE)

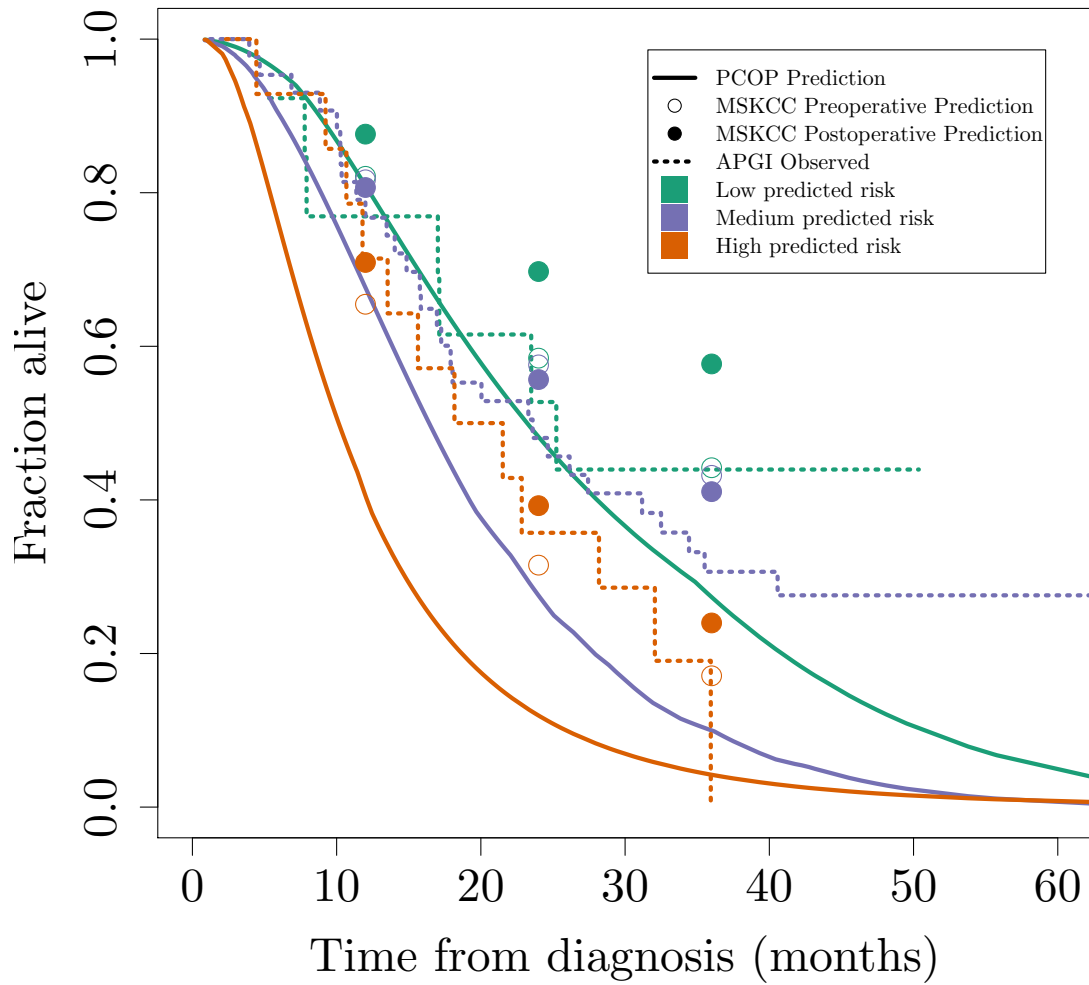
temp.km = survfit(Surv(data.apgi$Time, data.apgi$DSD) ~ gg.groups.apgi, conf.int = 1-temp.alpha)
temp.km = data.frame(surv = temp.km$surv, group = rep(gsub(".*=", "", names(temp.km$strata)), temp.km$strata))
temp.pred = summary(fit.gg, newdata = data.apgi, ci = FALSE)
temp.pred.times = temp.pred[[1]][,1]
```

```

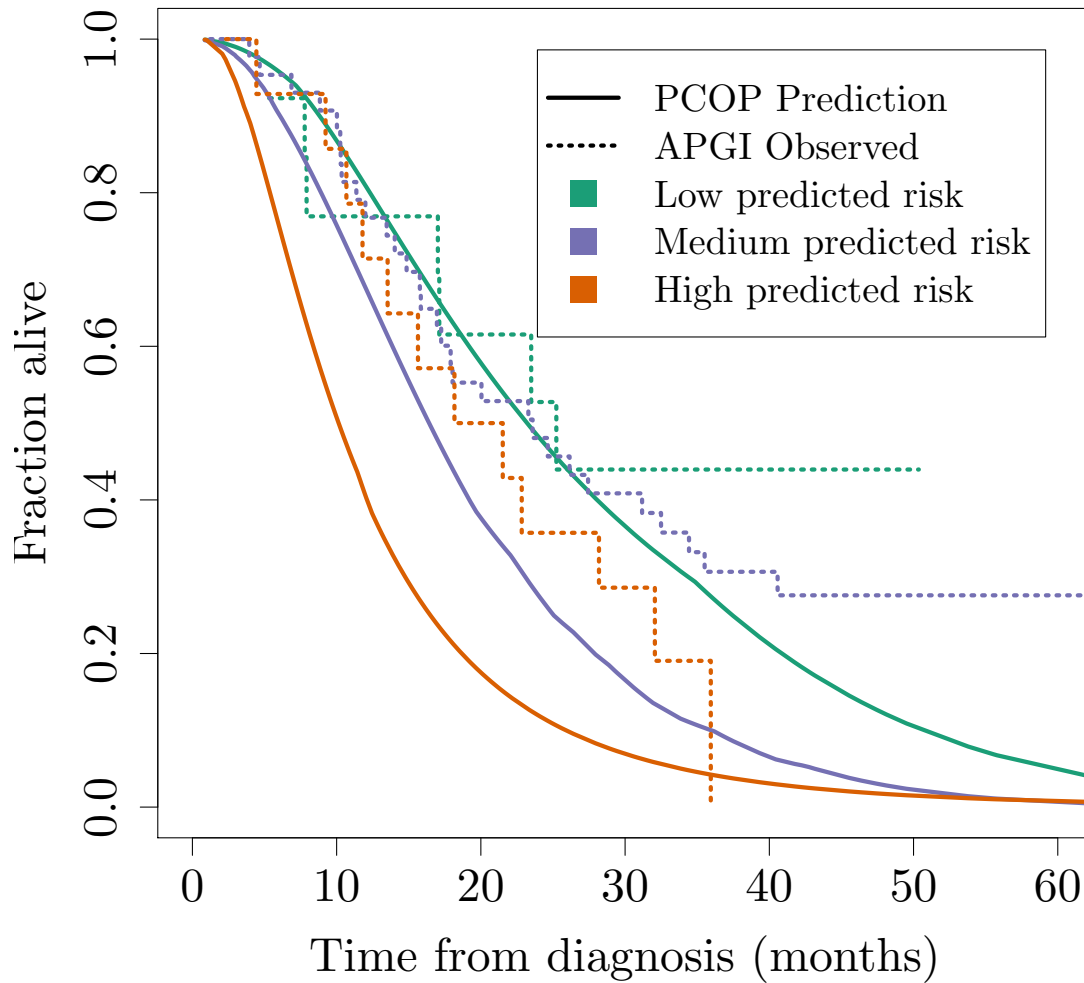
temp.pred.ests = sapply(temp.pred, function(x) x[,2])
temp.pred.ests = tapply(1:ncol(temp.pred.ests), gg.groups.apgi, function(is) apply(temp.pred.ests[,is],
temp.pred.lower = sapply(temp.pred.ests, function(x) x[1,])
temp.pred.meds = sapply(temp.pred.ests, function(x) x[2,])
temp.pred.upper = sapply(temp.pred.ests, function(x) x[3,])
temp.pred = data.frame(surv = as.vector(temp.pred.meds), group = rep(colnames(temp.pred.meds), each = n
temp.data = rbind(temp.km, temp.pred)
temp.data$time = temp.data$time / 365.25 * 12
temp.predpre.12mo = simplify2array(tapply(mskcc_pre.12mo.apgi, mskcc_pre.groups.apgi, quantile, probs =
temp.predpre.24mo = simplify2array(tapply(mskcc_pre.24mo.apgi, mskcc_pre.groups.apgi, quantile, probs =
temp.predpre.36mo = simplify2array(tapply(mskcc_pre.36mo.apgi, mskcc_pre.groups.apgi, quantile, probs =
temp.predpost.12mo = simplify2array(tapply(mskcc_post.12mo.apgi, mskcc_post.groups.apgi, quantile, probs =
temp.predpost.24mo = simplify2array(tapply(mskcc_post.24mo.apgi, mskcc_post.groups.apgi, quantile, probs =
temp.predpost.36mo = simplify2array(tapply(mskcc_post.36mo.apgi, mskcc_post.groups.apgi, quantile, probs =
temp.data2 = data.frame(
  surv = c(temp.predpre.12mo[2,], temp.predpre.24mo[2,], temp.predpre.36mo[2,], temp.predpost.12mo
  group = factor(rep(sort(unique(mskcc_pre.groups.apgi)), 6)),
  time = rep(c(12, 24, 36), each = 3),
  upper = c(temp.predpre.12mo[3,], temp.predpre.24mo[3,], temp.predpre.36mo[3,], temp.predpost.12mo
  lower = c(temp.predpre.12mo[1,], temp.predpre.24mo[1,], temp.predpre.36mo[1,], temp.predpost.12mo
  est = rep(c("MSKCC Preoperative", "MSKCC Postoperative"), each = 9))

plot(0 ~ 0, type = "n", xlim = c(0, 5*12), ylim = c(0, 1), xlab = "Time from diagnosis (months)", ylab =
temp.pal = brewer.pal(length(unique(gg.groups.apgi)), "Dark2")[c(1, 3, 2)]
names(temp.pal) = sort(unique(gg.groups.apgi))
for (temp.i in factor(sort(unique(gg.groups.apgi))))
{
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$
  points(surv ~ time, temp.data2[as.character(temp.data2$group) == as.character(temp.i) & temp.dat
  points(surv ~ time, temp.data2[as.character(temp.data2$group) == as.character(temp.i) & temp.dat
}
legend("topright", inset = 0.05, legend = c("PCOP Prediction", "MSKCC Preoperative Prediction", "MSKCC P

```



```
plot(0 ~ 0, type = "n", xlim = c(0, 5*12), ylim = c(0, 1), xlab = "Time from diagnosis (months)", ylab = "Fraction alive")
temp.pal = brewer.pal(length(unique(gg.groups.apgi)), "Dark2")[c(1, 3, 2)]
names(temp.pal) = sort(unique(gg.groups.apgi))
for (temp.i in factor(sort(unique(gg.groups.apgi))))
{
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$apgi == "Low predicted risk",])
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$apgi == "Medium predicted risk",])
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$apgi == "High predicted risk",])
}
legend("topright", inset = 0.05, legend = c("PCOP Prediction", "APCI Observed", "Low predicted risk", "Medium predicted risk", "High predicted risk"))
```



```
summary(coxph(Surv(data.apgi$Time, data.apgi$DSD) ~ factor(gg.groups.apgi)))

## Call:
## coxph(formula = Surv(data.apgi$Time, data.apgi$DSD) ~ factor(gg.groups.apgi))
##
## n= 73, number of events= 50
## (2 observations deleted due to missingness)
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## factor(gg.groups.apgi)2 0.182    1.199   0.421  0.43   0.67
## factor(gg.groups.apgi)3 0.584    1.793   0.477  1.22   0.22
##
##               exp(coef) exp(-coef) lower .95 upper .95
## factor(gg.groups.apgi)2    1.20    0.834   0.525   2.74
## factor(gg.groups.apgi)3    1.79    0.558   0.704   4.56
##
## Concordance= 0.533 (se = 0.039 )
## Rsquare= 0.024 (max possible= 0.993 )
## Likelihood ratio test= 1.79 on 2 df,  p=0.409
## Wald test               = 1.89 on 2 df,  p=0.389
## Score (logrank) test = 1.92 on 2 df,  p=0.383

summary(coxph(Surv(data.apgi$Time, data.apgi$DSD) ~ factor(mskcc_pre.groups.apgi)))
```



```
## Call:
## coxph(formula = Surv(data.apgi$Time, data.apgi$DSD) ~ factor(mskcc_pre.groups.apgi))
##
## n= 74, number of events= 50
## (1 observation deleted due to missingness)
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## factor(mskcc_pre.groups.apgi)2 -0.412    0.662    0.367 -1.12    0.26
## factor(mskcc_pre.groups.apgi)3 -0.058    0.944    0.449 -0.13    0.90
##
##               exp(coef) exp(-coef) lower .95 upper .95
## factor(mskcc_pre.groups.apgi)2    0.662    1.51    0.322    1.36
## factor(mskcc_pre.groups.apgi)3    0.944    1.06    0.392    2.27
##
## Concordance= 0.559 (se = 0.037 )
## Rsquare= 0.023 (max possible= 0.993 )
## Likelihood ratio test= 1.7 on 2 df, p=0.428
## Wald test = 1.75 on 2 df, p=0.417
## Score (logrank) test = 1.77 on 2 df, p=0.412

summary(coxph(Surv(data.apgi$Time, data.apgi$DSD) ~ factor(mskcc_post.groups.apgi)))

## Call:
## coxph(formula = Surv(data.apgi$Time, data.apgi$DSD) ~ factor(mskcc_post.groups.apgi))
##
## n= 74, number of events= 51
## (1 observation deleted due to missingness)
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## factor(mskcc_post.groups.apgi)2 1.526    4.598    0.531 2.87    0.0041
## factor(mskcc_post.groups.apgi)3 1.812    6.125    0.576 3.15    0.0016
##
##               exp(coef) exp(-coef) lower .95 upper .95
## factor(mskcc_post.groups.apgi)2    4.60    0.217    1.62    13.0
## factor(mskcc_post.groups.apgi)3    6.12    0.163    1.98    18.9
##
## Concordance= 0.624 (se = 0.04 )
## Rsquare= 0.184 (max possible= 0.993 )
## Likelihood ratio test= 15.1 on 2 df, p=0.000539
## Wald test = 10.1 on 2 df, p=0.00628
## Score (logrank) test = 12.3 on 2 df, p=0.00208
```

```
mskcc_pre.groups.dresden = cut(mskcc_pre.linpred.dresden, quantile(mskcc_pre.linpred.dresden, group_quantiles),
mskcc_post.groups.dresden = cut(mskcc_post.linpred.dresden, quantile(mskcc_post.linpred.dresden, group_quantiles),
gg.groups.dresden = cut(gg.linpred.dresden, quantile(gg.linpred.dresden, group_quantiles), labels = FALSE)

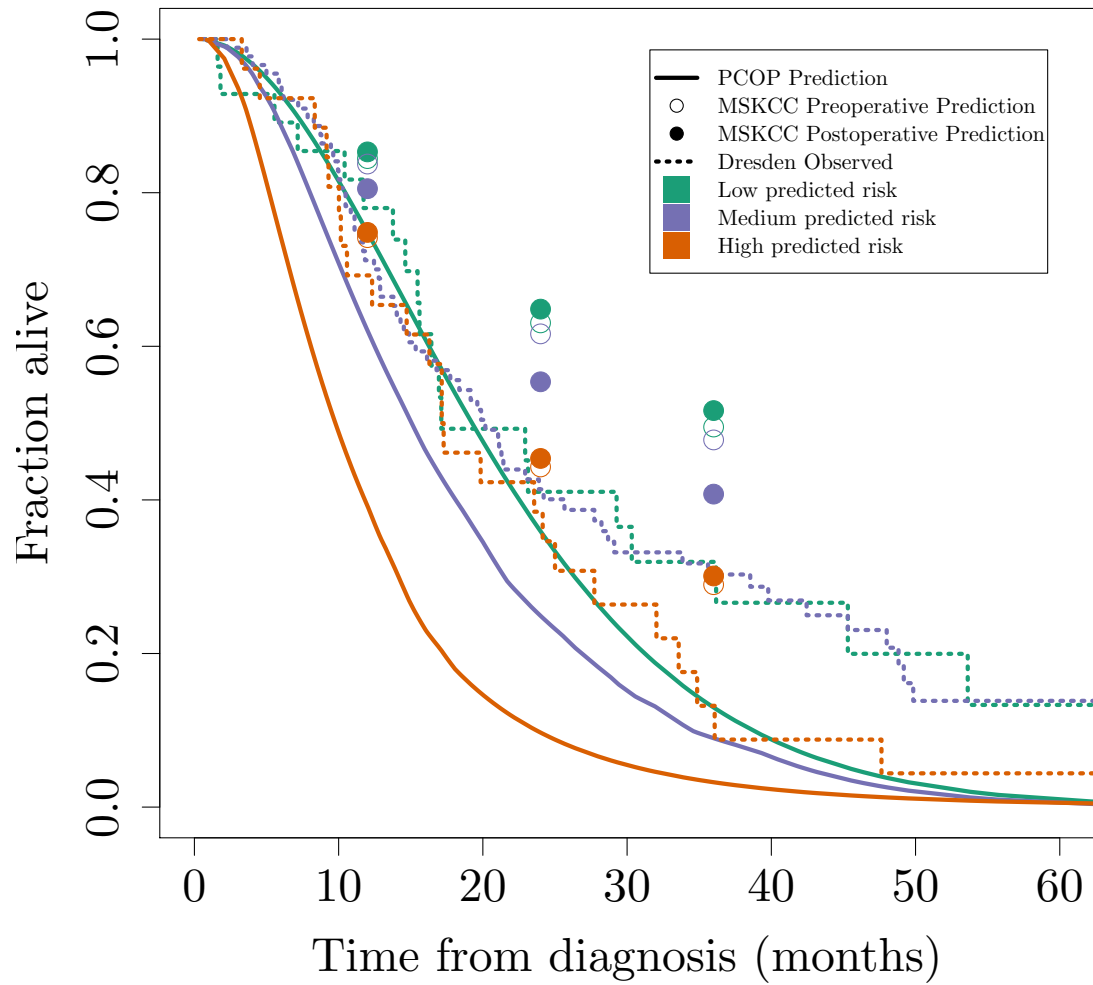
temp.km = survfit(Surv(data.dresden$Time, data.dresden$DSD) ~ gg.groups.dresden, conf.int = 1-temp.alpha)
temp.km = data.frame(surv = temp.km$surv, group = rep(gsub(".*=", "", names(temp.km$strata)), temp.km$strata))
temp.pred = summary(fit.gg, newdata = data.dresden, ci = FALSE)
temp.pred.times = temp.pred[[1]][,1]
temp.pred.ests = sapply(temp.pred, function(x) x[,2])
temp.pred.ests = tapply(1:ncol(temp.pred.ests), gg.groups.dresden, function(is) apply(temp.pred.ests[,is], 1, FUN = function(x) x[is,]))
temp.pred.lower = sapply(temp.pred.ests, function(x) x[1,])
```

```

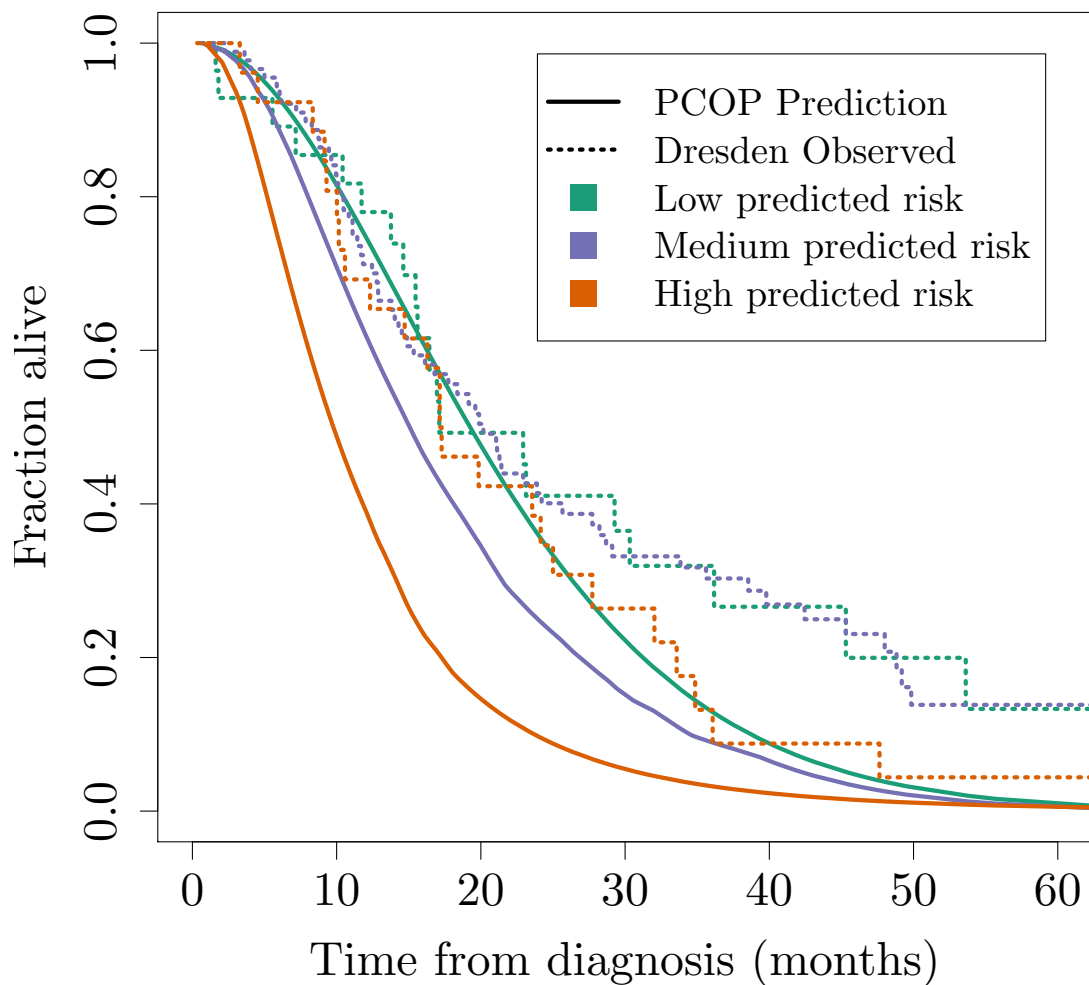
temp.pred.meds = sapply(temp.pred.ests, function(x) x[2,])
temp.pred.upper = sapply(temp.pred.ests, function(x) x[3,])
temp.pred = data.frame(surv = as.vector(temp.pred.meds), group = rep(colnames(temp.pred.meds), each = nrow(temp.pred.meds)))
temp.data = rbind(temp.km, temp.pred)
temp.data$time = temp.data$time / 365.25 * 12
temp.predpre.12mo = simplify2array(tapply(mskcc_pre.12mo.dresden, mskcc_pre.groups.dresden, quantile, probs = probs, na.rm = TRUE))
temp.predpre.24mo = simplify2array(tapply(mskcc_pre.24mo.dresden, mskcc_pre.groups.dresden, quantile, probs = probs, na.rm = TRUE))
temp.predpre.36mo = simplify2array(tapply(mskcc_pre.36mo.dresden, mskcc_pre.groups.dresden, quantile, probs = probs, na.rm = TRUE))
temp.predpost.12mo = simplify2array(tapply(mskcc_post.12mo.dresden, mskcc_post.groups.dresden, quantile, probs = probs, na.rm = TRUE))
temp.predpost.24mo = simplify2array(tapply(mskcc_post.24mo.dresden, mskcc_post.groups.dresden, quantile, probs = probs, na.rm = TRUE))
temp.predpost.36mo = simplify2array(tapply(mskcc_post.36mo.dresden, mskcc_post.groups.dresden, quantile, probs = probs, na.rm = TRUE))
temp.data2 = data.frame(
  surv = c(temp.predpre.12mo[2,], temp.predpre.24mo[2,], temp.predpre.36mo[2,], temp.predpost.12mo[2,], temp.predpost.24mo[2,], temp.predpost.36mo[2,]),
  group = factor(rep(sort(unique(mskcc_pre.groups.dresden)), 6)),
  time = rep(c(12, 24, 36), each = 3),
  upper = c(temp.predpre.12mo[3,], temp.predpre.24mo[3,], temp.predpre.36mo[3,], temp.predpost.12mo[3,], temp.predpost.24mo[3,], temp.predpost.36mo[3,]),
  lower = c(temp.predpre.12mo[1,], temp.predpre.24mo[1,], temp.predpre.36mo[1,], temp.predpost.12mo[1,], temp.predpost.24mo[1,], temp.predpost.36mo[1,]),
  est = rep(c("MSKCC Preoperative", "MSKCC Postoperative"), each = 9))

plot(0 ~ 0, type = "n", xlim = c(0, 5*12), ylim = c(0, 1), xlab = "Time from diagnosis (months)", ylab = "PCOP Prediction")
temp.pal = brewer.pal(length(unique(gg.groups.dresden)), "Dark2")[c(1, 3, 2)]
names(temp.pal) = sort(unique(gg.groups.dresden))
for (temp.i in factor(sort(unique(gg.groups.dresden))))
{
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$est == "MSKCC Preoperative"], col = temp.pal[temp.i], lty = 1)
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$est == "MSKCC Postoperative"], col = temp.pal[temp.i], lty = 2)
  points(surv ~ time, temp.data2[as.character(temp.data2$group) == as.character(temp.i) & temp.data2$est == "MSKCC Preoperative"], col = temp.pal[temp.i], lty = 1)
  points(surv ~ time, temp.data2[as.character(temp.data2$group) == as.character(temp.i) & temp.data2$est == "MSKCC Postoperative"], col = temp.pal[temp.i], lty = 2)
}
legend("topright", inset = 0.05, legend = c("PCOP Prediction", "MSKCC Preoperative Prediction", "MSKCC Postoperative Prediction"), col = c("black", temp.pal[1], temp.pal[2]), lty = c(1, 1, 2))

```



```
plot(0 ~ 0, type = "n", xlim = c(0, 5*12), ylim = c(0, 1), xlab = "Time from diagnosis (months)", ylab = "Fraction alive")
temp.pal = brewer.pal(length(unique(gg.groups.dresden)), "Dark2")[c(1, 3, 2)]
names(temp.pal) = sort(unique(gg.groups.dresden))
for (temp.i in factor(sort(unique(gg.groups.dresden))))
{
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$dresden == temp.i,])
  lines(surv ~ time, temp.data[as.character(temp.data$group) == as.character(temp.i) & temp.data$dresden == temp.i,])
}
legend("topright", inset = 0.05, legend = c("PCOP Prediction", "Dresden Observed", "Low predicted risk", "Medium predicted risk", "High predicted risk"))
```



```
summary(coxph(Surv(data.dresden$Time, data.dresden$DSD) ~ factor(gg.groups.dresden)))
```

## Call:  
## coxph(formula = Surv(data.dresden\$Time, data.dresden\$DSD) ~ factor(gg.groups.dresden))  
##  
## n= 149, number of events= 111  
## (1 observation deleted due to missingness)  
##  
##

	coef	exp(coef)	se(coef)	z	Pr(> z )
## factor(gg.groups.dresden)2	0.0305	1.0310	0.2555	0.12	0.90
## factor(gg.groups.dresden)3	0.3364	1.4000	0.3038	1.11	0.27

```
##  
##
```

	exp(coef)	exp(-coef)	lower .95	upper .95
## factor(gg.groups.dresden)2	1.03	0.970	0.625	1.70
## factor(gg.groups.dresden)3	1.40	0.714	0.772	2.54

```
##  
## Concordance= 0.52 (se = 0.027 )  
## Rsquare= 0.012 (max possible= 0.998 )  
## Likelihood ratio test= 1.73 on 2 df, p=0.421  
## Wald test = 1.84 on 2 df, p=0.399  
## Score (logrank) test = 1.85 on 2 df, p=0.397  
  
summary(coxph(Surv(data.dresden$Time, data.dresden$DSD) ~ factor(mskcc_pre.groups.dresden)))
```

```
## Call:
## coxph(formula = Surv(data.dresden$Time, data.dresden$DSD) ~ factor(mskcc_pre.groups.dresden))
##
## n= 149, number of events= 112
## (1 observation deleted due to missingness)
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## factor(mskcc_pre.groups.dresden)2 0.0797    1.0830    0.2483 0.32    0.75
## factor(mskcc_pre.groups.dresden)3 0.3448    1.4117    0.2938 1.17    0.24
##
##               exp(coef) exp(-coef) lower .95 upper .95
## factor(mskcc_pre.groups.dresden)2      1.08      0.923    0.666    1.76
## factor(mskcc_pre.groups.dresden)3      1.41      0.708    0.794    2.51
##
## Concordance= 0.517 (se = 0.028 )
## Rsquare= 0.01 (max possible= 0.998 )
## Likelihood ratio test= 1.57 on 2 df, p=0.456
## Wald test              = 1.64 on 2 df, p=0.441
## Score (logrank) test = 1.65 on 2 df, p=0.438

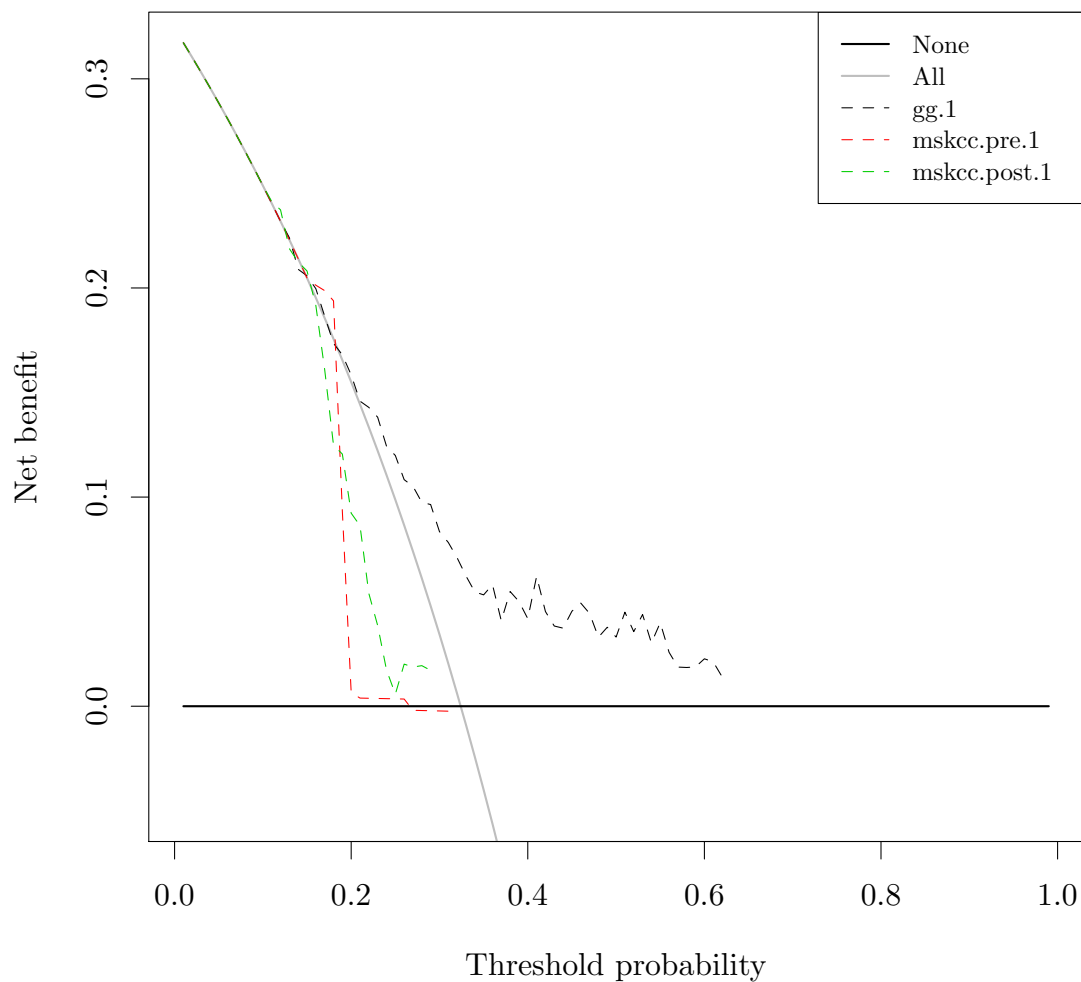
summary(coxph(Surv(data.dresden$Time, data.dresden$DSD) ~ factor(mskcc_post.groups.dresden)))

## Call:
## coxph(formula = Surv(data.dresden$Time, data.dresden$DSD) ~ factor(mskcc_post.groups.dresden))
##
## n= 149, number of events= 111
## (1 observation deleted due to missingness)
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## factor(mskcc_post.groups.dresden)2 0.431    1.539    0.284 1.51    0.1298
## factor(mskcc_post.groups.dresden)3 1.019    2.771    0.334 3.05    0.0023
##
##               exp(coef) exp(-coef) lower .95
## factor(mskcc_post.groups.dresden)2      1.54      0.650    0.881
## factor(mskcc_post.groups.dresden)3      2.77      0.361    1.439
##
##               upper .95
## factor(mskcc_post.groups.dresden)2      2.69
## factor(mskcc_post.groups.dresden)3      5.34
##
## Concordance= 0.569 (se = 0.027 )
## Rsquare= 0.063 (max possible= 0.998 )
## Likelihood ratio test= 9.73 on 2 df, p=0.00772
## Wald test              = 10.1 on 2 df, p=0.00648
## Score (logrank) test = 10.5 on 2 df, p=0.0052
```

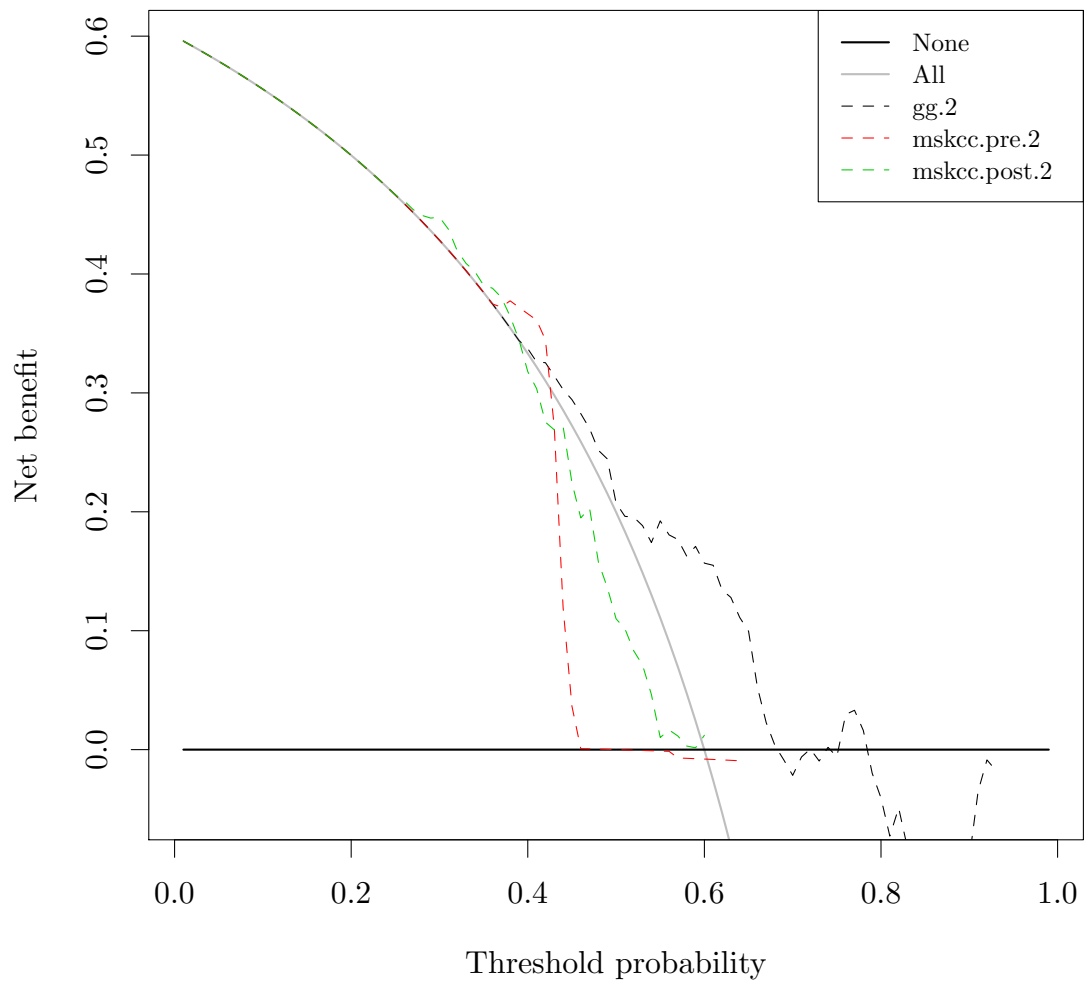
Decision curve analysis.

```
source("stdca.R")
temp.data = data.frame(Time = data.glasgow$Time, DSD = data.glasgow$DSD*1,
  gg.1 = 1-gg.prob.glasgow[val.prob.times == 365,], gg.2 = 1-gg.prob.glasgow[val.prob.times == 365*2,],
  mskcc.pre.1 = 1-mskcc_pre.12mo.glasgow, mskcc.pre.2 = 1-mskcc_pre.24mo.glasgow, mskcc.pre.3 = 1-mskcc_pre.36mo.glasgow,
  mskcc.post.1 = 1-mskcc_post.12mo.glasgow, mskcc.post.2 = 1-mskcc_post.24mo.glasgow, mskcc.post.3 = 1-mskcc_post.36mo.glasgow)
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.1", "mskcc.pre.1", "mskcc.pre.2", "mskcc.pre.3", "mskcc.post.1", "mskcc.post.2", "mskcc.post.3")))
## [1] "gg.1: No observations with risk greater than 63% that have followup through the timepoint select"
```

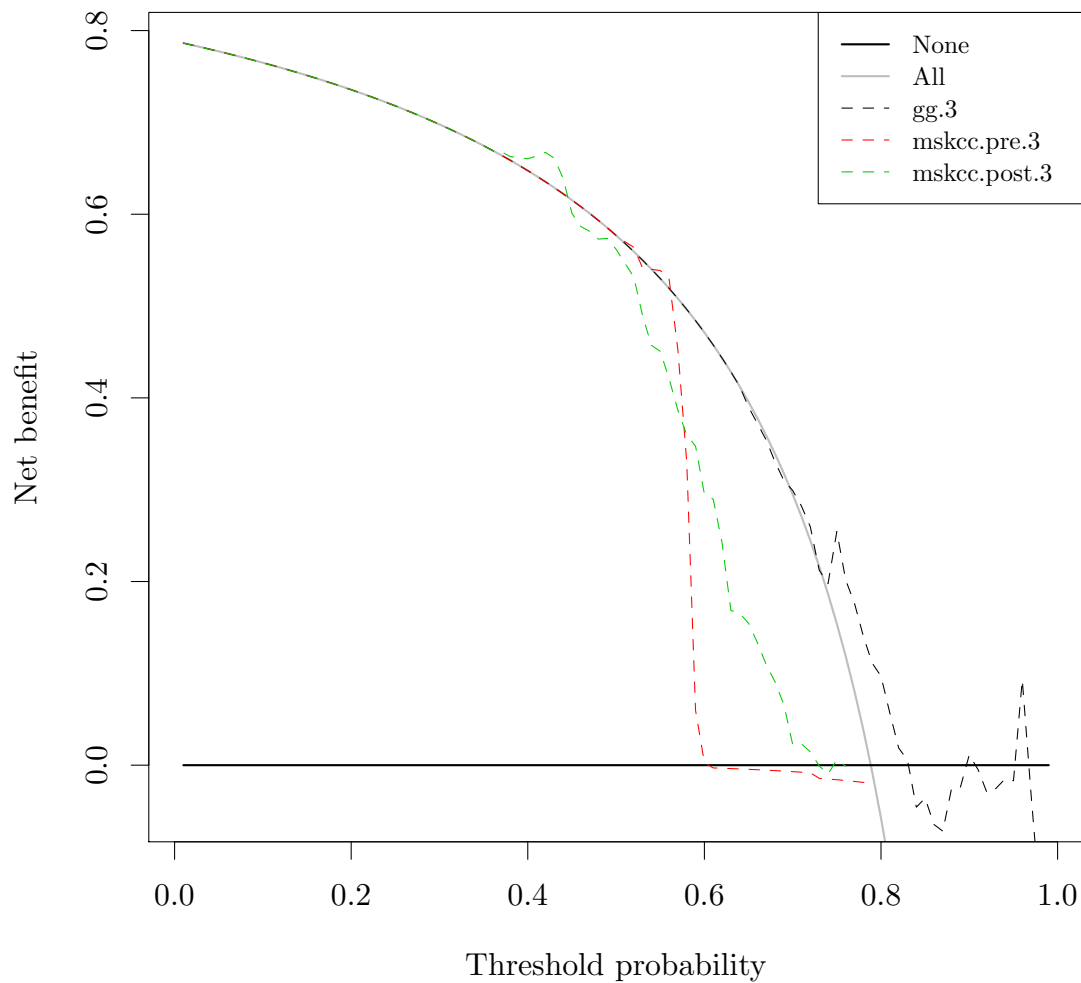
```
## [2] "mskcc.pre.1: No observations with risk greater than 32%, and therefore net benefit not calculabl
## [3] "mskcc.post.1: No observations with risk greater than 30% that have followup through the timepoint
```



```
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.2", "mskcc.pre
## [1] "gg.2: No observations with risk greater than 94% that have followup through the timepoint select
## [2] "mskcc.pre.2: No observations with risk greater than 65%, and therefore net benefit not calculabl
## [3] "mskcc.post.2: No observations with risk greater than 61% that have followup through the timepoint
```



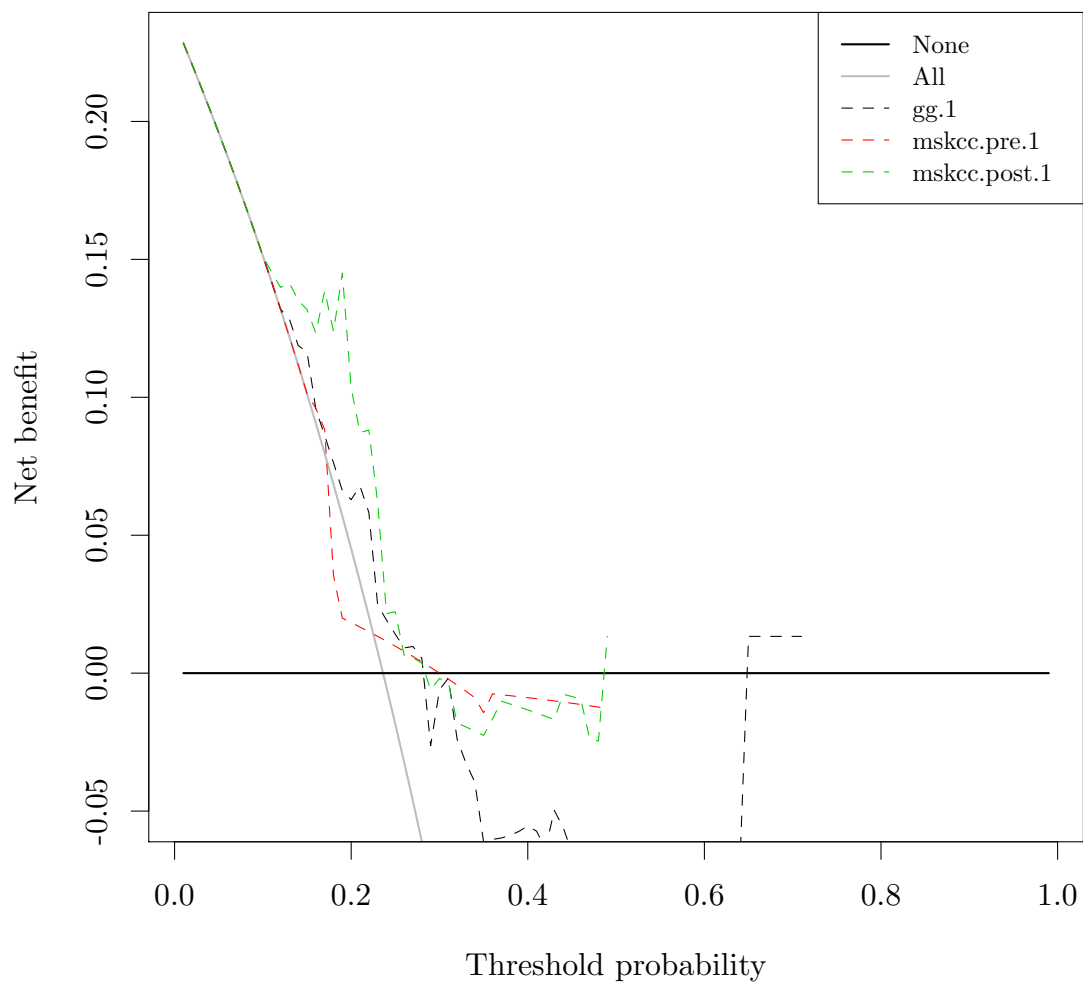
```
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.3", "mskcc.pre.3", "mskcc.post.3")))
## [1] "mskcc.pre.3: No observations with risk greater than 80%, and therefore net benefit not calculabl"
## [2] "mskcc.post.3: No observations with risk greater than 77% that have followup through the timepoint"
```



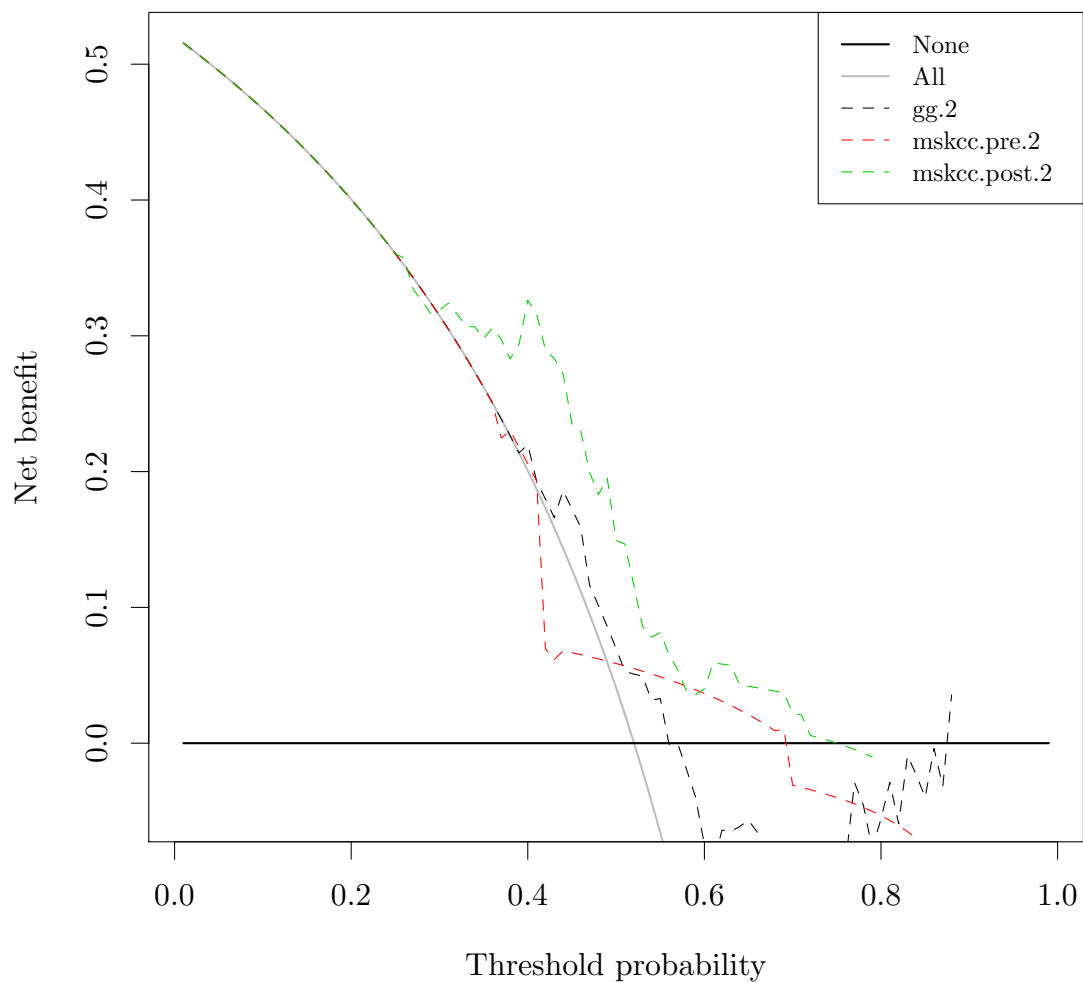
```
temp.data = data.frame(Time = data.apgi$Time, DSD = data.apgi$DSD*1,
  gg.1 = 1-gg.prob.apgi[val.prob.times == 365,], gg.2 = 1-gg.prob.apgi[val.prob.times == 365*2,], gg.3 = 1-gg.prob.apgi[val.prob.times == 365*3,],
  mskcc.pre.1 = 1-mskcc_pre.12mo.apgi, mskcc.pre.2 = 1-mskcc_pre.24mo.apgi, mskcc.pre.3 = 1-mskcc_pre.36mo.apgi,
  mskcc.post.1 = 1-mskcc_post.12mo.apgi, mskcc.post.2 = 1-mskcc_post.24mo.apgi, mskcc.post.3 = 1-mskcc_post.36mo.apgi,
  invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.1", "mskcc.pre.1", "mskcc.pre.2", "mskcc.pre.3",
  "mskcc.post.1", "mskcc.post.2", "mskcc.post.3"))))

## [1] "gg.1: No observations with risk greater than 72%, and therefore net benefit not calculable in the threshold range [0.72, 1.0]"
## [2] "mskcc.pre.1: No observations with risk greater than 50%, and therefore net benefit not calculable in the threshold range [0.5, 1.0]"
## [3] "mskcc.post.1: No observations with risk greater than 50%, and therefore net benefit not calculable in the threshold range [0.5, 1.0]"
```

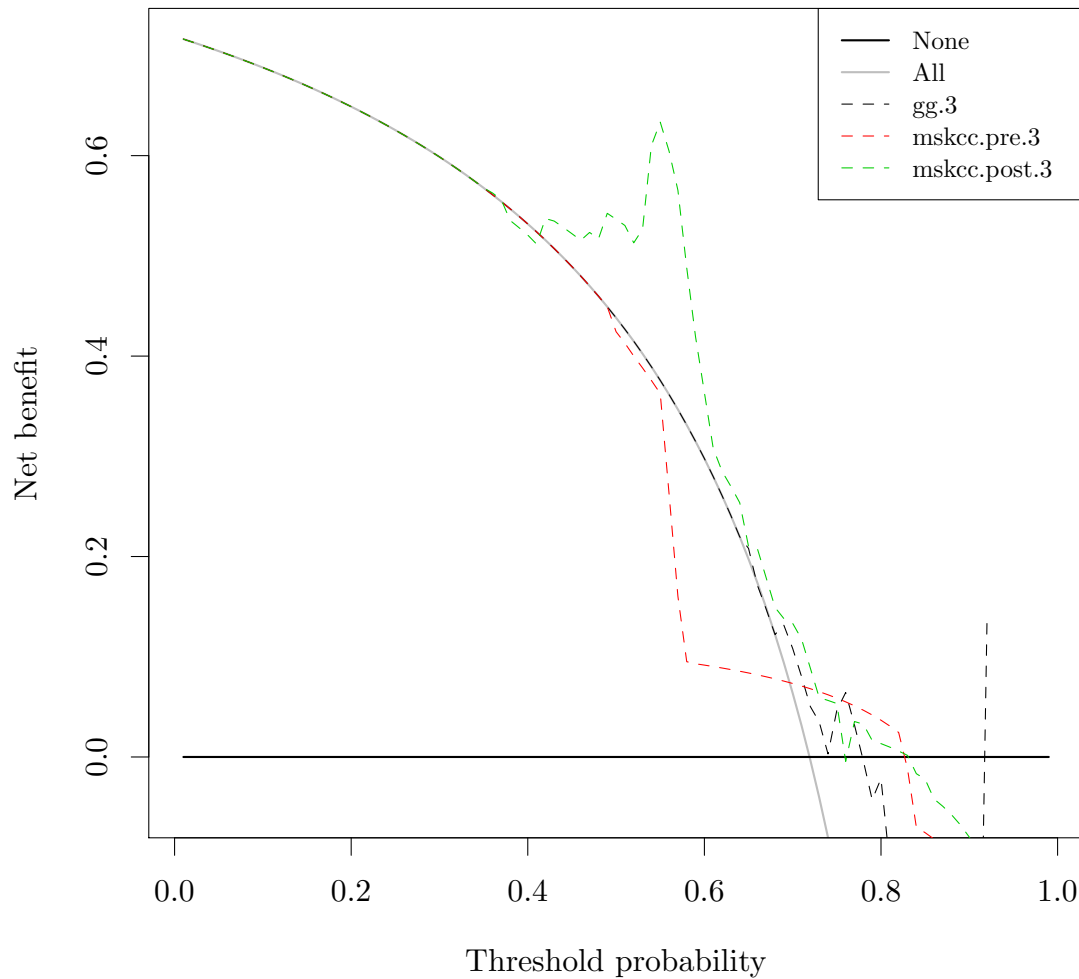




```
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.2", "mskcc.pre.2", "mskcc.post.2")))
## [1] "gg.2: No observations with risk greater than 89% that have followup through the timepoint selected"
## [2] "mskcc.pre.2: No observations with risk greater than 85%, and therefore net benefit not calculable"
## [3] "mskcc.post.2: No observations with risk greater than 80% that have followup through the timepoint selected"
```

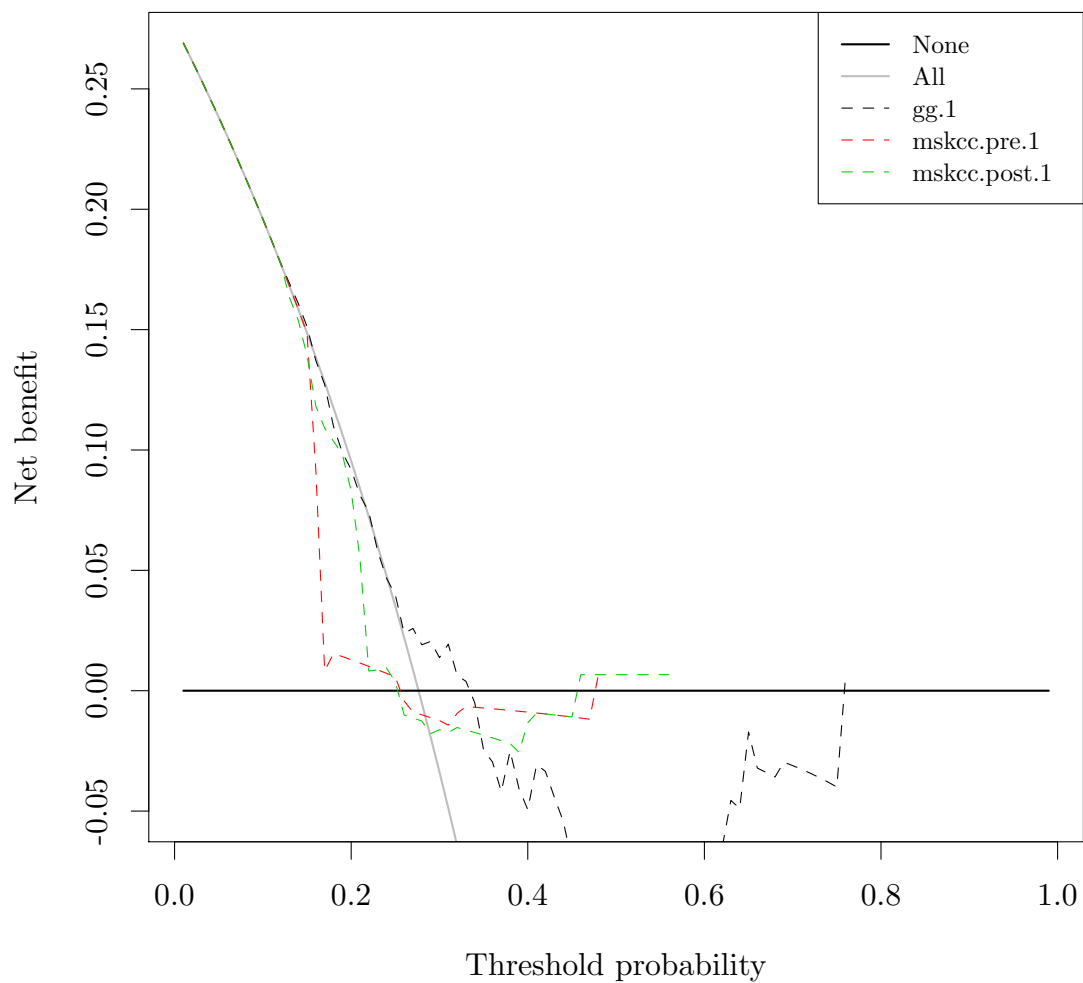


```
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.3", "mskcc.pre.3", "mskcc.post.3")))
## [1] "gg.3: No observations with risk greater than 93% that have followup through the timepoint selected"
## [2] "mskcc.pre.3: No observations with risk greater than 95%, and therefore net benefit not calculable"
## [3] "mskcc.post.3: No observations with risk greater than 92% that have followup through the timepoint selected"
```

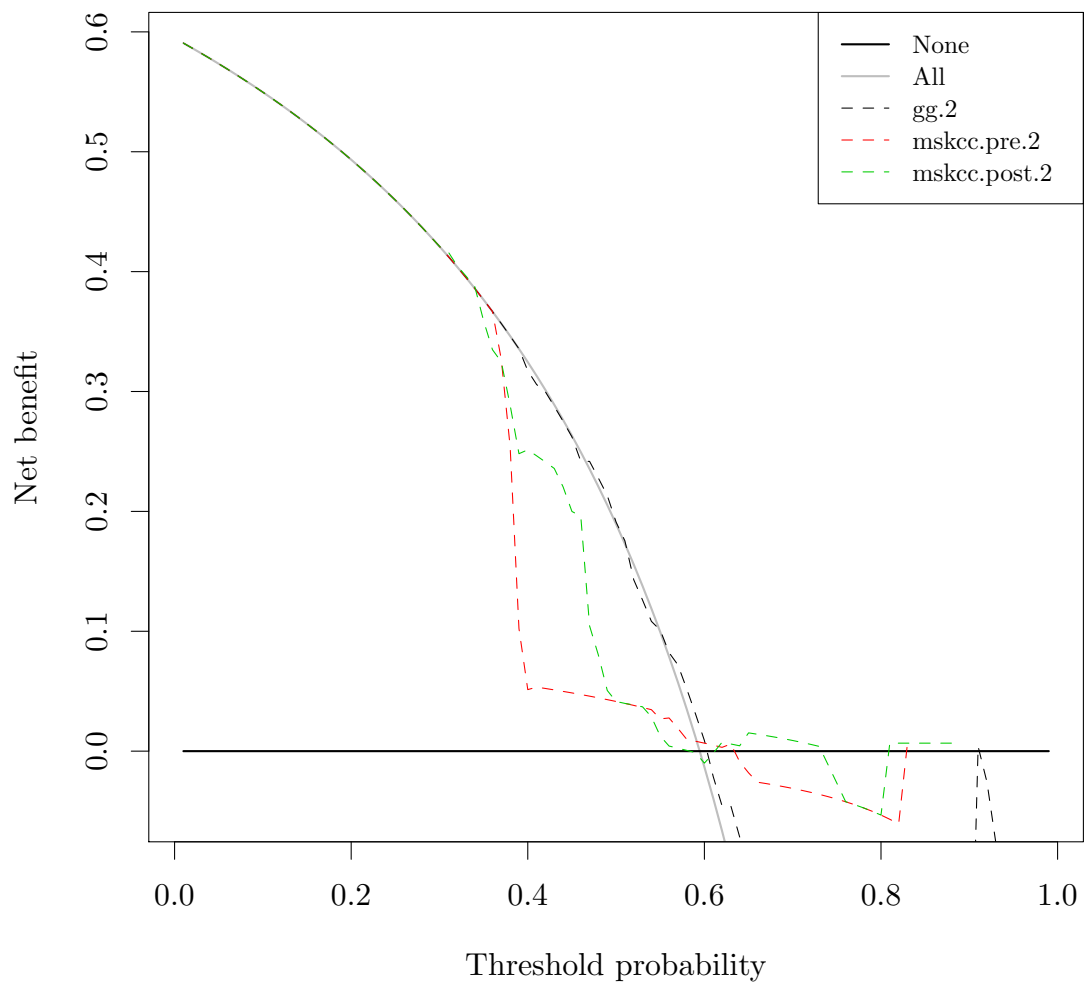


```
temp.data = data.frame(Time = data.dresden$Time, DSD = data.dresden$DSD*1,
  gg.1 = 1-gg.prob.dresden[val.prob.times == 365,], gg.2 = 1-gg.prob.dresden[val.prob.times == 365*2,],
  mskcc.pre.1 = 1-mskcc_pre.12mo.dresden, mskcc.pre.2 = 1-mskcc_pre.24mo.dresden, mskcc.pre.3 = 1-mskcc_pre.36mo.dresden,
  mskcc.post.1 = 1-mskcc_post.12mo.dresden, mskcc.post.2 = 1-mskcc_post.24mo.dresden, mskcc.post.3 = 1-mskcc_post.36mo.dresden)
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.1", "mskcc.pre.1", "mskcc.pre.2", "mskcc.pre.3",
  "mskcc.post.1", "mskcc.post.2", "mskcc.post.3")))

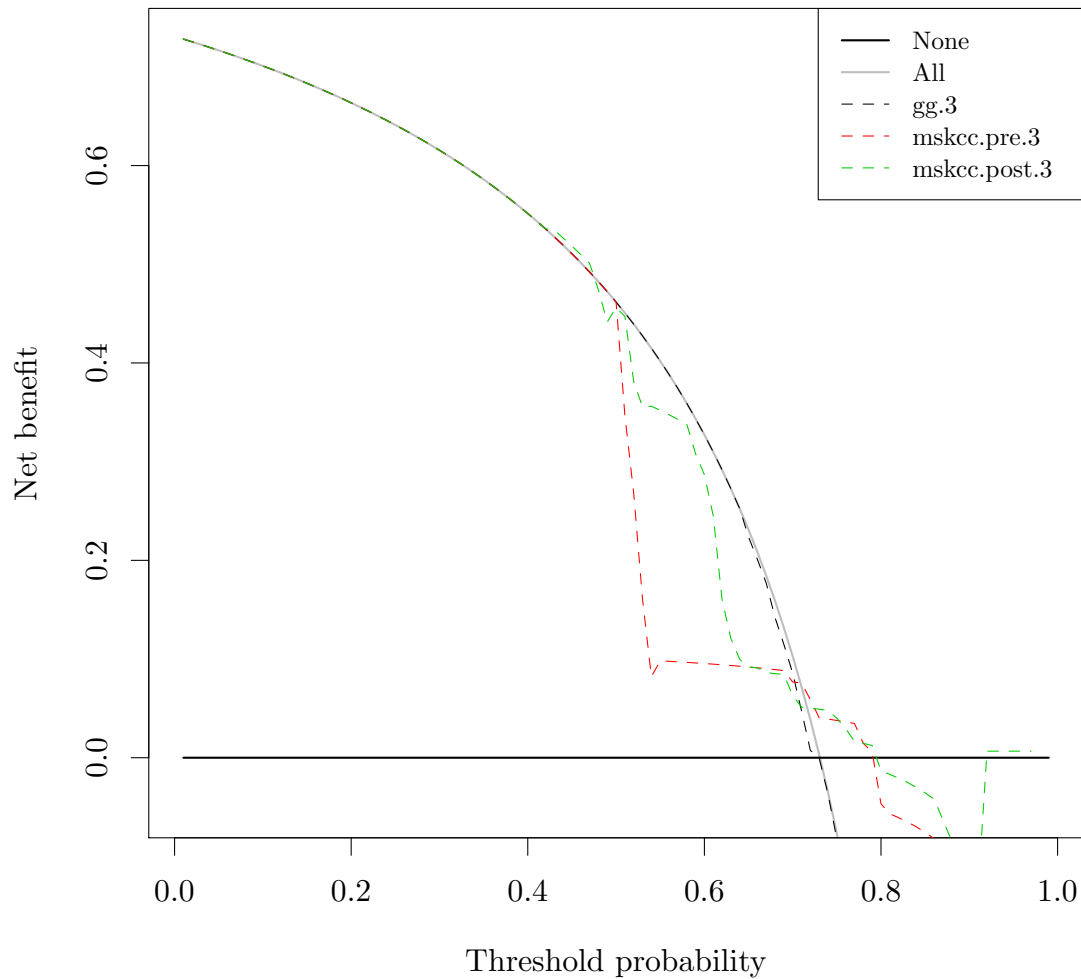
## [1] "gg.1: No observations with risk greater than 77%, and therefore net benefit not calculable in the threshold range [0.77, 1.0]"
## [2] "mskcc.pre.1: No observations with risk greater than 49%, and therefore net benefit not calculable in the threshold range [0.49, 1.0]"
## [3] "mskcc.post.1: No observations with risk greater than 57%, and therefore net benefit not calculable in the threshold range [0.57, 1.0]"
```



```
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.2", "mskcc.pre.2", "mskcc.post.2")))
## [1] "gg.2: No observations with risk greater than 94% that have followup through the timepoint selected"
## [2] "mskcc.pre.2: No observations with risk greater than 84%, and therefore net benefit not calculable"
## [3] "mskcc.post.2: No observations with risk greater than 90%, and therefore net benefit not calculable"
```



```
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.3", "mskcc.pre.3", "mskcc.post.3")))
## [1] "mskcc.pre.3: No observations with risk greater than 94%, and therefore net benefit not calculable"
## [2] "mskcc.post.3: No observations with risk greater than 98%, and therefore net benefit not calculable"
```



## 4.6 Brier score

```
calcIBS = function(surv, pred, pred_times, max_time, min_time = 0)
{
  stopifnot(nrow(surv) == nrow(pred) && length(pred_times) == ncol(pred))

  n = nrow(surv)
  marg_survfit = survfit(surv ~ 1)
  marg_censfit = survfit(Surv(surv[,1], !surv[,2]) ~ 1)
  marg_surv_func = approxfun(marg_survfit$time, marg_survfit$surv, method = "constant", yleft = 1, yright = 0)
  marg_cens_func = approxfun(marg_censfit$time, marg_censfit$surv, method = "constant", yleft = 1, yright = 0)

  pred_funcs = apply(pred, 1, function(pat_preds) approxfun(pred_times, pat_preds, yleft = 1, yright = 0))

  indiv_patient_bsc = function(pat_i, tstars)
  {
    observed_time = surv[pat_i, 1]
    observed_event = surv[pat_i, 2]
    pred_func = pred_funcs[[pat_i]]
    category = 1*(observed_time <= tstars & observed_event) + 2*(observed_time > tstars) + 3*(observed_time < min_time)
    bsc = rep(NA, length(tstars))
    bsc[category == 1] = pred_func(tstars[category == 1])^2 / marg_cens_func(observed_time)
  }
}
```

```

        bsc[category == 2] = (1 - pred_func(tstars[category == 2]))^2 / marg_cens_func(tstars[category == 2])
        bsc[category == 3] = 0
    }
    bsc

    bsc_func = function(tstars) { rowMeans(sapply(1:n, function(pat_i) indiv_patient_bsc(pat_i, tstars))) }

    weight_func = function(tstars) { (1 - marg_surv_func(tstars)) / (1 - marg_surv_func(max_time)) }

    # Be slack and do trapezoidal int. with a fine grid. It should be possible
    # to calculate the int. exactly but I cbfed.
    int_grid = seq(min_time, max_time, length.out = 1e3)
    bsc_vals = bsc_func(int_grid)
    weight_vals = weight_func(int_grid)
    int_vals = bsc_vals * weight_vals
    ibsc = (2*sum(int_vals) - int_vals[1] - int_vals[length(int_vals)]) * (diff(range(int_grid))) /

    return(list(bsc = bsc_vals, weights = weight_vals, eval_times = int_grid, ibsc = ibsc))
}

calcBSsingle = function(surv, pred, pred_time)
{
    n = nrow(surv)
    obs_time = surv[,1]
    obs_event = surv[,2]
    marg_censfit = survfit(Surv(obs_time, !obs_event) ~ 1)
    marg_cens_func = approxfun(marg_censfit$time, marg_censfit$surv, method = "constant", yleft = 1, yright = 0)

    brier_val = rep(NA, n)
    cat = 1*I(obs_time <= pred_time & obs_event) + 2*I(obs_time > pred_time) + 3*I(obs_time <= pred_time & !obs_event)
    brier_val[cat == 1] = (pred[cat == 1])^2 / marg_cens_func(obs_time[cat == 1])
    brier_val[cat == 2] = (1-pred[cat == 2])^2 / marg_cens_func(pred_time)
    brier_val[cat == 3] = 0

    mean(brier_val)
}

```

```

mskcc_post.12mo.glasgow.brier = calcBSsingle(Surv(data.glasgow$Time, data.glasgow$DSD), mskcc_post.12mo.glasgow.brier, mskcc_post.12mo.glasgow.brier)
mskcc_post.24mo.glasgow.brier = calcBSsingle(Surv(data.glasgow$Time, data.glasgow$DSD), mskcc_post.24mo.glasgow.brier, mskcc_post.24mo.glasgow.brier)
mskcc_post.36mo.glasgow.brier = calcBSsingle(Surv(data.glasgow$Time, data.glasgow$DSD), mskcc_post.36mo.glasgow.brier, mskcc_post.36mo.glasgow.brier)
mskcc_pre.12mo.glasgow.brier = calcBSsingle(Surv(data.glasgow$Time, data.glasgow$DSD), mskcc_pre.12mo.glasgow.brier, mskcc_pre.12mo.glasgow.brier)
mskcc_pre.24mo.glasgow.brier = calcBSsingle(Surv(data.glasgow$Time, data.glasgow$DSD), mskcc_pre.24mo.glasgow.brier, mskcc_pre.24mo.glasgow.brier)
mskcc_pre.36mo.glasgow.brier = calcBSsingle(Surv(data.glasgow$Time, data.glasgow$DSD), mskcc_pre.36mo.glasgow.brier, mskcc_pre.36mo.glasgow.brier)
gg.path.glasgow.brier = calcIBS(Surv(data.glasgow$Time, data.glasgow$DSD), t(sapply(gg.path.glasgow, function(x) calcBSsingle(Surv(x$Time, x$DSD), x$brier, x$brier))))
km0.path.glasgow.brier = calcIBS(Surv(data.glasgow$Time, data.glasgow$DSD), matrix(fit.km0$surv, nrow = ncol(fit.km0$surv)))

```

```

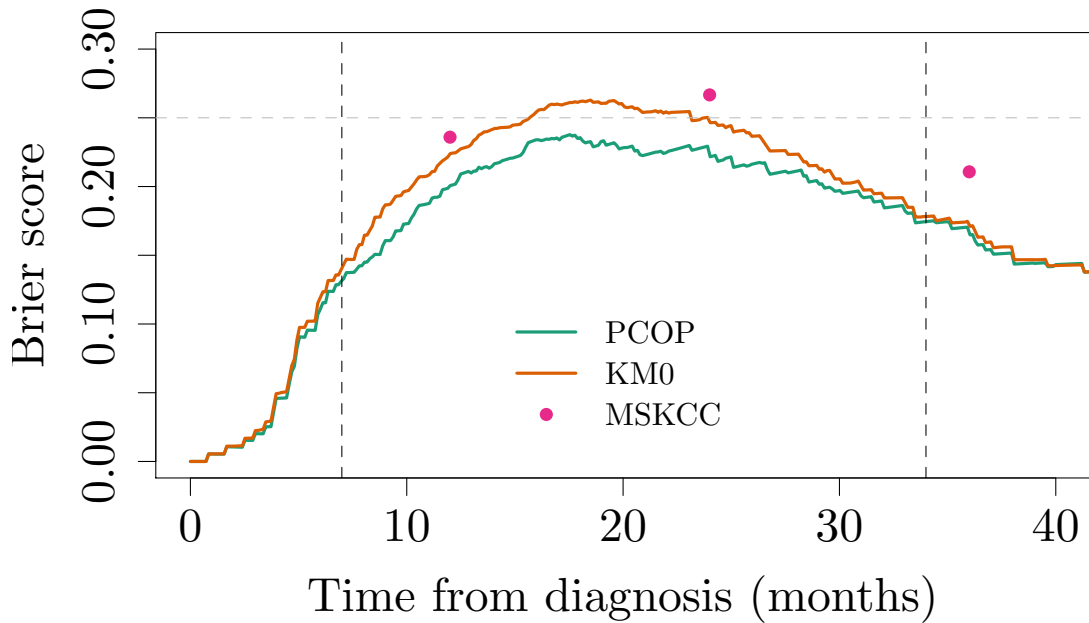
mskcc_post.12mo.apgi.brier = calcBSsingle(Surv(data.apgi$Time, data.apgi$DSD), mskcc_post.12mo.apgi.brier, mskcc_post.12mo.apgi.brier)
mskcc_post.24mo.apgi.brier = calcBSsingle(Surv(data.apgi$Time, data.apgi$DSD), mskcc_post.24mo.apgi.brier, mskcc_post.24mo.apgi.brier)
mskcc_post.36mo.apgi.brier = calcBSsingle(Surv(data.apgi$Time, data.apgi$DSD), mskcc_post.36mo.apgi.brier, mskcc_post.36mo.apgi.brier)
mskcc_pre.12mo.apgi.brier = calcBSsingle(Surv(data.apgi$Time, data.apgi$DSD), mskcc_pre.12mo.apgi.brier, mskcc_pre.12mo.apgi.brier)
mskcc_pre.24mo.apgi.brier = calcBSsingle(Surv(data.apgi$Time, data.apgi$DSD), mskcc_pre.24mo.apgi.brier, mskcc_pre.24mo.apgi.brier)
mskcc_pre.36mo.apgi.brier = calcBSsingle(Surv(data.apgi$Time, data.apgi$DSD), mskcc_pre.36mo.apgi.brier, mskcc_pre.36mo.apgi.brier)

```

```
gg.path.apgi.brier = calcIBS(Surv(data.apgi$Time, data.apgi$DSD), t(sapply(gg.path.apgi, function(x) x[,
km0.path.apgi.brier = calcIBS(Surv(data.apgi$Time, data.apgi$DSD), matrix(fit.km0$surv, nrow = nrow(data
```

```
mskcc_post.12mo.dresden.brier = calcBSsingle(Surv(data.dresden$Time, data.dresden$DSD), mskcc_post.12mo.d
mskcc_post.24mo.dresden.brier = calcBSsingle(Surv(data.dresden$Time, data.dresden$DSD), mskcc_post.24mo
mskcc_post.36mo.dresden.brier = calcBSsingle(Surv(data.dresden$Time, data.dresden$DSD), mskcc_post.36mo
mskcc_pre.12mo.dresden.brier = calcBSsingle(Surv(data.dresden$Time, data.dresden$DSD), mskcc_pre.12mo.d
mskcc_pre.24mo.dresden.brier = calcBSsingle(Surv(data.dresden$Time, data.dresden$DSD), mskcc_pre.24mo.d
mskcc_pre.36mo.dresden.brier = calcBSsingle(Surv(data.dresden$Time, data.dresden$DSD), mskcc_pre.36mo.d
gg.path.dresden.brier = calcIBS(Surv(data.dresden$Time, data.dresden$DSD), t(sapply(gg.path.dresden, fun
km0.path.dresden.brier = calcIBS(Surv(data.dresden$Time, data.dresden$DSD), matrix(fit.km0$surv, nrow =
```

```
plot(gg.path.glasgow.brier$eval_times/365.25*12, gg.path.glasgow.brier$bsc, col = pal["gg"], type = "l",
lines(gg.path.glasgow.brier$eval_times/365.25*12, km0.path.glasgow.brier$bsc, col = pal["km0"], lwd = 3),
points(c(12, 24, 36), c(mskcc_pre.12mo.glasgow.brier, mskcc_pre.24mo.glasgow.brier, mskcc_pre.36mo.glas
abline(h = 0.25, col = "grey", lty = "dashed")
abline(v = c(7, 34), lty = "dashed")
legend("bottom",
      legend = c(      "PCOP",              "KM0",          "MSKCC"),
      pch = c(      NA,              NA,              16),
      col = c(      pal["gg"],          pal["km0"], pal["mskcc.pre"]),
      lty = c(      "solid",          "solid",          NA),
      inset = 0.05, lwd = 3, bty = "n")
```



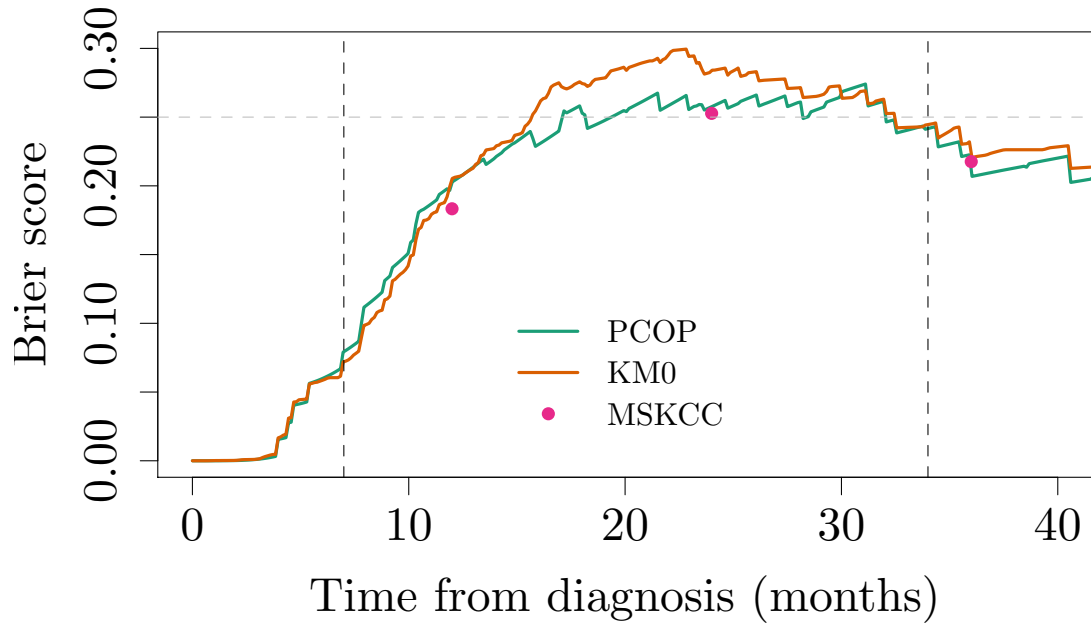
```
plot(gg.path.apgi.brier$eval_times/365.25*12, gg.path.apgi.brier$bsc, col = pal["gg"], type = "l", ylim
lines(gg.path.apgi.brier$eval_times/365.25*12, km0.path.apgi.brier$bsc, col = pal["km0"], lwd = 3)
points(c(12, 24, 36), c(mskcc_pre.12mo.apgi.brier, mskcc_pre.24mo.apgi.brier, mskcc_pre.36mo.apgi.brier)
abline(h = 0.25, col = "grey", lty = "dashed")
abline(v = c(7, 34), lty = "dashed")
legend("bottom",
      legend = c(      "PCOP",              "KM0",          "MSKCC"),
```



```

pch = c(      NA,                      NA,      16),
col = c(      pal["gg"],                pal["km0"], pal["mskcc.pre"]),
lty = c(      "solid",                  "solid",  NA),
inset = 0.05, lwd = 3, bty = "n")

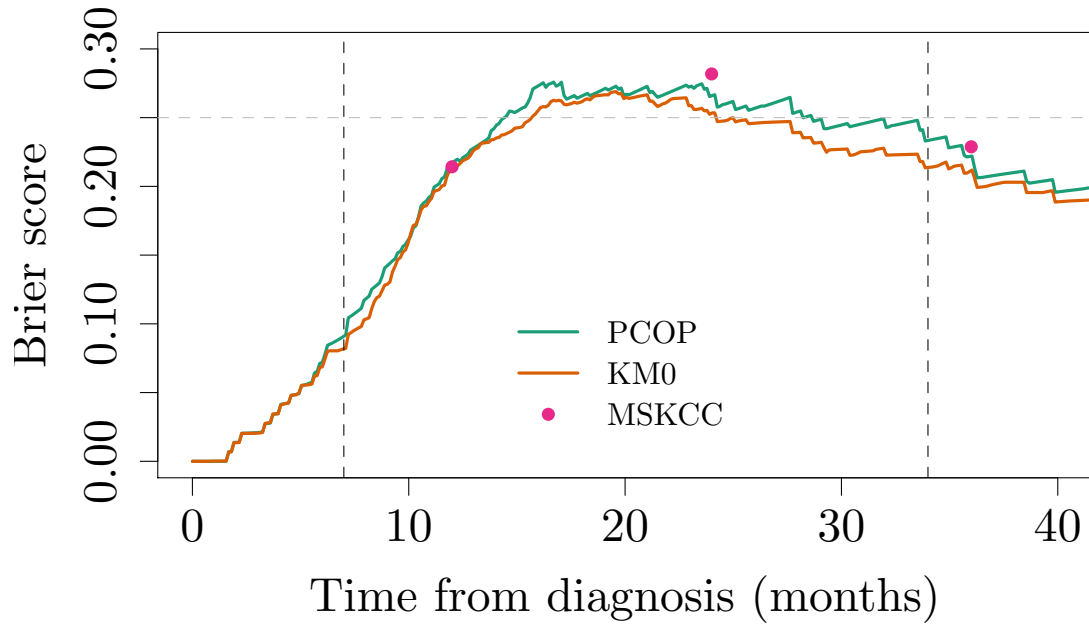
```



```

plot(gg.path.dresden.brier$eval_times/365.25*12, gg.path.dresden.brier$bsc, col = pal["gg"], type = "l",
lines(gg.path.dresden.brier$eval_times/365.25*12, km0.path.dresden.brier$bsc, col = pal["km0"], lwd = 3),
points(c(12, 24, 36), c(mskcc_pre.12mo.dresden.brier, mskcc_pre.24mo.dresden.brier, mskcc_pre.36mo.dresden.brier),
abline(h = 0.25, col = "grey", lty = "dashed")
abline(v = c(7, 34), lty = "dashed")
legend("bottom",
      legend = c(      "PCOP",                "KM0",      "MSKCC"),
pch = c(      NA,                      NA,      16),
col = c(      pal["gg"],                pal["km0"], pal["mskcc.pre"]),
lty = c(      "solid",                  "solid",  NA),
inset = 0.05, lwd = 3, bty = "n")

```



```

probs_bs_boot_func_glasgow = function(d, i) {
  bs.mskcc.postop.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_post.12mo.glasgow[i], 12/12*365.25)
  bs.mskcc.postop.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_post.24mo.glasgow[i], 24/12*365.25)
  bs.mskcc.postop.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_post.36mo.glasgow[i], 36/12*365.25)
  bs.mskcc.preop.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_pre.12mo.glasgow[i], 12/12*365.25)
  bs.mskcc.preop.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_pre.24mo.glasgow[i], 24/12*365.25)
  bs.mskcc.preop.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_pre.36mo.glasgow[i], 36/12*365.25)

  bs.gg.vals = t(sapply(gg.path.glasgow[i], function(path) approx(path[,1], path[,2], c(12, 24, 36))))
  rownames(bs.gg.vals) <- NULL
  bs.gg.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), bs.gg.vals[,1], 12/12*365.25)
  bs.gg.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), bs.gg.vals[,2], 24/12*365.25)
  bs.gg.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), bs.gg.vals[,3], 36/12*365.25)

  bs.km0.vals = approx(fit.km0$time, fit.km0$surv, c(12, 24, 36)/12*365.25)$y
  bs.km0.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), rep(bs.km0.vals[1], nrow(d[i])), 12/12*365.25)
  bs.km0.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), rep(bs.km0.vals[2], nrow(d[i])), 24/12*365.25)
  bs.km0.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), rep(bs.km0.vals[3], nrow(d[i])), 36/12*365.25)

  result = c(
    bs.gg.12 - bs.km0.12,          bs.mskcc.preop.12 - bs.km0.12,
    bs.gg.12 - bs.mskcc.preop.12,  bs.mskcc.preop.24 - bs.km0.24,
    bs.gg.24 - bs.km0.24,          bs.mskcc.preop.24 - bs.km0.24,
    bs.gg.24 - bs.mskcc.preop.24,  bs.mskcc.preop.36 - bs.km0.36,
    bs.gg.36 - bs.km0.36,          bs.mskcc.preop.36 - bs.km0.36,
    bs.gg.36 - bs.mskcc.preop.36)

  names(result) <- NULL
  result
}

set.seed(20150208)
deltaBrier.boot.glasgow = boot(data.glasgow, probs_bs_boot_func_glasgow, R = 500)

```

```

deltaBrier.boot.glasgow.cis = t(sapply(1:ncol(deltaBrier.boot.glasgow$t), function(i) boot.ci(deltaBrier
colnames(deltaBrier.boot.glasgow.cis) = c("level", "lowindex", "highindex", "lci", "uci")
rownames(deltaBrier.boot.glasgow.cis) = c(
  "12:gg-km0", "12:pre-km0", "12:gg-pre",
  "24:gg-km0", "24:pre-km0", "24:gg-pre",
  "36:gg-km0", "36:pre-km0", "36:gg-pre")
deltaBrier.boot.glasgow

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data.glasgow, statistic = probs_bs_boot_func_glasgow,
##       R = 500)
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* -0.023252 -5.591e-04   0.011020
## t2*  0.012000  5.097e-04   0.014791
## t3* -0.035252 -1.069e-03   0.018703
## t4* -0.024707 -1.173e-03   0.011163
## t5*  0.020378  1.780e-04   0.020822
## t6* -0.045085 -1.351e-03   0.022651
## t7* -0.006137 -3.073e-04   0.006092
## t8*  0.039775 -9.123e-06   0.018277
## t9* -0.045912 -2.982e-04   0.018448

deltaBrier.boot.glasgow.cis

##           level lowindex highindex      lci      uci
## 12:gg-km0    0.95    19.36    493.3 -0.0438016  0.0001641
## 12:pre-km0    0.95    10.07    485.4 -0.0179132  0.0401415
## 12:gg-pre     0.95     9.88    485.4 -0.0753277 -0.0035136
## 24:gg-km0    0.95    17.35    492.2 -0.0471870 -0.0023731
## 24:pre-km0    0.95    11.87    487.8 -0.0189747  0.0617515
## 24:gg-pre     0.95    19.24    493.3 -0.0845755  0.0024417
## 36:gg-km0    0.95    15.48    490.9 -0.0174246  0.0056702
## 36:pre-km0    0.95     7.75    482.0  0.0002576  0.0703455
## 36:gg-pre     0.95    17.88    492.7 -0.0791661 -0.0078058

```

```

probs_bs_boot_func_apgi = function(d, i) {
  bs.mskcc.postop.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_post.12mo.apgi[i], 12/12*365.25)
  bs.mskcc.postop.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_post.24mo.apgi[i], 24/12*365.25)
  bs.mskcc.postop.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_post.36mo.apgi[i], 36/12*365.25)
  bs.mskcc.preop.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_pre.12mo.apgi[i], 12/12*365.25)
  bs.mskcc.preop.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_pre.24mo.apgi[i], 24/12*365.25)
  bs.mskcc.preop.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_pre.36mo.apgi[i], 36/12*365.25)

  bs.gg.vals = t(sapply(gg.path.apgi[i], function(path) approx(path[,1], path[,2], c(12, 24, 36)/12)
rownames(bs.gg.vals) <- NULL
  bs.gg.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), bs.gg.vals[,1], 12/12*365.25)

```

```

bs.gg.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), bs.gg.vals[,2], 24/12*365.25)
bs.gg.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), bs.gg.vals[,3], 36/12*365.25)

bs.km0.vals = approx(fit.km0$time, fit.km0$surv, c(12, 24, 36)/12*365.25)$y
bs.km0.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), rep(bs.km0.vals[1], nrow(d[i,])), 12/12*365.25)
bs.km0.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), rep(bs.km0.vals[2], nrow(d[i,])), 24/12*365.25)
bs.km0.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), rep(bs.km0.vals[3], nrow(d[i,])), 36/12*365.25)

result = c(
  bs.gg.12 - bs.km0.12,          bs.mskcc.preop.12 - bs.km0.12,
  bs.gg.12 - bs.mskcc.preop.12,
  bs.gg.24 - bs.km0.24,          bs.mskcc.preop.24 - bs.km0.24,
  bs.gg.24 - bs.mskcc.preop.24,
  bs.gg.36 - bs.km0.36,          bs.mskcc.preop.36 - bs.km0.36,
  bs.gg.36 - bs.mskcc.preop.36)

names(result) <- NULL
result
}

set.seed(20150208)
deltaBrier.boot.apgi = boot(data.apgi, probs_bs_boot_func_apgi, R = 500)
deltaBrier.boot.apgi.cis = t(sapply(1:ncol(deltaBrier.boot.apgi$t), function(i) boot.ci(deltaBrier.boot.apgi, i)))
colnames(deltaBrier.boot.apgi.cis) = c("level", "lowindex", "highindex", "lci", "uci")
rownames(deltaBrier.boot.apgi.cis) = c(
  "12:gg-km0", "12:pre-km0", "12:gg-pre",
  "24:gg-km0", "24:pre-km0", "24:gg-pre",
  "36:gg-km0", "36:pre-km0", "36:gg-pre")
deltaBrier.boot.apgi

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data.apgi, statistic = probs_bs_boot_func_apgi, R = 500)
##
##
## Bootstrap Statistics :
##      original      bias      std. error
## t1* -0.002467 -9.427e-04    0.01533
## t2* -0.021902 -9.299e-04    0.01871
## t3*  0.019435 -1.284e-05    0.02357
## t4* -0.026163  1.455e-04    0.01626
## t5* -0.031015 -2.957e-03    0.03088
## t6*  0.004852  3.102e-03    0.03210
## t7* -0.013158  5.419e-04    0.01076
## t8* -0.002300 -1.978e-03    0.03104
## t9* -0.010858  2.520e-03    0.03157

deltaBrier.boot.apgi.cis

##           level lowindex highindex      lci      uci
## 12:gg-km0   0.95   19.66    493.6 -0.03018 0.028926
## 12:pre-km0   0.95   19.61    493.7 -0.05458 0.021352

```

```
## 12:gg-pre 0.95 9.26 484.6 -0.03157 0.063344
## 24:gg-km0 0.95 11.80 487.7 -0.05845 0.005853
## 24:pre-km0 0.95 24.25 495.0 -0.08215 0.036547
## 24:gg-pre 0.95 9.20 484.1 -0.05843 0.066835
## 36:gg-km0 0.95 7.17 481.3 -0.03799 0.005808
## 36:pre-km0 0.95 13.87 489.6 -0.06168 0.053566
## 36:gg-pre 0.95 14.34 490.2 -0.06126 0.052894
```

```
probs_bs_boot_func_dresden = function(d, i) {
  bs.mskcc.postop.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_post.12mo.dresden[i], 12/12*365.25)
  bs.mskcc.postop.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_post.24mo.dresden[i], 24/12*365.25)
  bs.mskcc.postop.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_post.36mo.dresden[i], 36/12*365.25)
  bs.mskcc.preop.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_pre.12mo.dresden[i], 12/12*365.25)
  bs.mskcc.preop.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_pre.24mo.dresden[i], 24/12*365.25)
  bs.mskcc.preop.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), mskcc_pre.36mo.dresden[i], 36/12*365.25)

  bs.gg.vals = t(sapply(gg.path.dresden[i], function(path) approx(path[,1], path[,2], c(12, 24, 36)/12*365.25)$y
rownames(bs.gg.vals) <- NULL
  bs.gg.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), bs.gg.vals[,1], 12/12*365.25)
  bs.gg.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), bs.gg.vals[,2], 24/12*365.25)
  bs.gg.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), bs.gg.vals[,3], 36/12*365.25)

  bs.km0.vals = approx(fit.km0$time, fit.km0$surv, c(12, 24, 36)/12*365.25)$y
  bs.km0.12 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), rep(bs.km0.vals[1], nrow(d[i,])), 12/12*365.25)
  bs.km0.24 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), rep(bs.km0.vals[2], nrow(d[i,])), 24/12*365.25)
  bs.km0.36 = calcBSsingle(Surv(d$Time[i], d$DSD[i]), rep(bs.km0.vals[3], nrow(d[i,])), 36/12*365.25)

  result = c(
    bs.gg.12 - bs.km0.12,
    bs.mskcc.preop.12 - bs.km0.12,
    bs.gg.12 - bs.mskcc.preop.12,
    bs.gg.24 - bs.km0.24,
    bs.mskcc.preop.24 - bs.km0.24,
    bs.gg.24 - bs.mskcc.preop.24,
    bs.gg.36 - bs.km0.36,
    bs.mskcc.preop.36 - bs.km0.36,
    bs.gg.36 - bs.mskcc.preop.36)

  names(result) <- NULL
  result
}

set.seed(20150208)
deltaBrier.boot.dresden = boot(data.dresden, probs_bs_boot_func_dresden, R = 500)
deltaBrier.boot.dresden.cis = t(sapply(1:ncol(deltaBrier.boot.dresden$t), function(i) boot.ci(deltaBrier.boot.dresden,
colnames(deltaBrier.boot.dresden.cis) = c("level", "lowindex", "highindex", "lci", "uci")
rownames(deltaBrier.boot.dresden.cis) = c(
  "12:gg-km0", "12:pre-km0", "12:gg-pre",
  "24:gg-km0", "24:pre-km0", "24:gg-pre",
  "36:gg-km0", "36:pre-km0", "36:gg-pre")
deltaBrier.boot.dresden

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
```

```
## Call:
## boot(data = data.dresden, statistic = probs_bs_boot_func_dresden,
##       R = 500)
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1*  0.0034421  9.367e-04   0.011108
## t2*  0.0004868  6.525e-05   0.016340
## t3*  0.0029552  8.714e-04   0.020299
## t4*  0.0126527  4.762e-04   0.010369
## t5*  0.0285863 -1.651e-04   0.026848
## t6* -0.0159336  6.414e-04   0.031273
## t7*  0.0103725  3.284e-04   0.006289
## t8*  0.0172025 -6.142e-04   0.025904
## t9* -0.0068300  9.426e-04   0.028929
```

```
deltaBrier.boot.dresden.cis
```

```
##           level lowindex highindex      lci      uci
## 12:gg-km0  0.95      8.37    482.4 -0.020654 0.02330
## 12:pre-km0  0.95     13.25    489.2 -0.028488 0.03549
## 12:gg-pre   0.95      6.20    478.0 -0.042748 0.03807
## 24:gg-km0  0.95     13.87    489.8 -0.007239 0.03688
## 24:pre-km0  0.95     10.10    485.6 -0.029593 0.07763
## 24:gg-pre   0.95     14.01    489.9 -0.075056 0.05172
## 36:gg-km0  0.95      9.38    484.3 -0.001999 0.02198
## 36:pre-km0  0.95     11.04    486.9 -0.037358 0.06442
## 36:gg-pre   0.95     10.83    486.4 -0.063253 0.05389
```

```
temp.time = gsub(".*", "", rownames(deltaBrier.boot.glasgow.cis))
temp.methodpos = gsub(".*", "", gsub(".*", "", rownames(deltaBrier.boot.glasgow.cis)))
temp.methodneg = gsub(".*-", "", rownames(deltaBrier.boot.glasgow.cis))
temp.methods = sort(unique(c(temp.methodpos, temp.methodneg)))
tapply(1:length(temp.time), temp.time, function(is) {
  res = matrix(0, nrow = length(temp.methods), ncol = length(temp.methods))
  rownames(res) = temp.methods
  colnames(res) = temp.methods
  # Make res signed. 0 => NS. +1 => row is better than col (BS_row - BS_col < 0). -1 => row is
  res[cbind(temp.methodpos[is], temp.methodneg[is])] = (sign(deltaBrier.boot.glasgow.cis[is, "uci"]
  res[cbind(temp.methodneg[is], temp.methodpos[is])] = (sign(deltaBrier.boot.glasgow.cis[is, "uci"]
  res
})
```

```
## $`12`
##      gg km0 pre
## gg    0  0  1
## km0   0  0  0
## pre  -1  0  0
##
## $`24`
##      gg km0 pre
## gg    0  1  0
## km0  -1  0  0
```

```
## pre 0 0 0
##
## $`36`
##      gg km0 pre
## gg    0 0 1
## km0   0 0 1
## pre  -1 -1 0
```

```
temp.time = gsub(".*", "", rownames(deltaBrier.boot.apgi.cis))
temp.methodpos = gsub(".*:", "", gsub("-.*", "", rownames(deltaBrier.boot.apgi.cis)))
temp.methodneg = gsub(".*-", "", rownames(deltaBrier.boot.apgi.cis))
temp.methods = sort(unique(c(temp.methodpos, temp.methodneg)))
tapply(1:length(temp.time), temp.time, function(is) {
  res = matrix(0, nrow = length(temp.methods), ncol = length(temp.methods))
  rownames(res) = temp.methods
  colnames(res) = temp.methods
  # Make res signed. 0 => NS. +1 => row is better than col (BS_row - BS_col < 0). -1 => row is
  res[cbind(temp.methodpos[is], temp.methodneg[is])] = (sign(deltaBrier.boot.apgi.cis[is, "uci"]))
  res[cbind(temp.methodneg[is], temp.methodpos[is])] = (sign(deltaBrier.boot.apgi.cis[is, "uci"]))
  res
})
```

```
## $`12`
##      gg km0 pre
## gg    0 0 0
## km0   0 0 0
## pre   0 0 0
##
## $`24`
##      gg km0 pre
## gg    0 0 0
## km0   0 0 0
## pre   0 0 0
##
## $`36`
##      gg km0 pre
## gg    0 0 0
## km0   0 0 0
## pre   0 0 0
```

```
temp.time = gsub(".*", "", rownames(deltaBrier.boot.dresden.cis))
temp.methodpos = gsub(".*:", "", gsub("-.*", "", rownames(deltaBrier.boot.dresden.cis)))
temp.methodneg = gsub(".*-", "", rownames(deltaBrier.boot.dresden.cis))
temp.methods = sort(unique(c(temp.methodpos, temp.methodneg)))
tapply(1:length(temp.time), temp.time, function(is) {
  res = matrix(0, nrow = length(temp.methods), ncol = length(temp.methods))
  rownames(res) = temp.methods
  colnames(res) = temp.methods
  # Make res signed. 0 => NS. +1 => row is better than col (BS_row - BS_col < 0). -1 => row is
  res[cbind(temp.methodpos[is], temp.methodneg[is])] = (sign(deltaBrier.boot.dresden.cis[is, "uci"]))
  res[cbind(temp.methodneg[is], temp.methodpos[is])] = (sign(deltaBrier.boot.dresden.cis[is, "uci"]))
  res
})
```

```

})

## $`12`
##      gg km0 pre
## gg    0   0   0
## km0    0   0   0
## pre    0   0   0
##
## $`24`
##      gg km0 pre
## gg    0   0   0
## km0    0   0   0
## pre    0   0   0
##
## $`36`
##      gg km0 pre
## gg    0   0   0
## km0    0   0   0
## pre    0   0   0

```

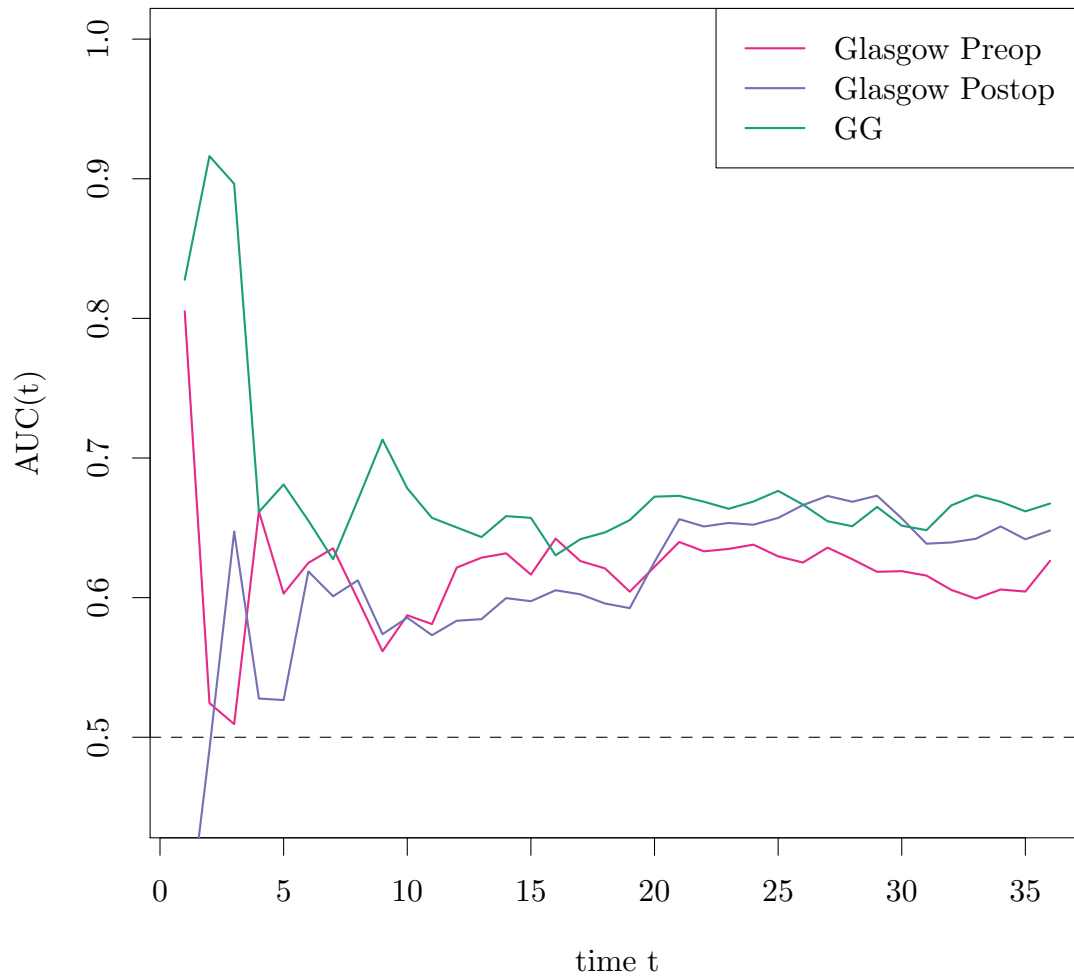
Cumulative-dynamic:

```

mskcc_pre.cdroc.glasgow = timeROC(data.glasgow$Time/365.25*12, data.glasgow$DSD, mskcc_pre.linpred.glasgow)
mskcc_post.cdroc.glasgow = timeROC(data.glasgow$Time/365.25*12, data.glasgow$DSD, mskcc_post.linpred.glasgow)
gg.cdroc.glasgow = timeROC(data.glasgow$Time/365.25*12, data.glasgow$DSD, gg.linpred.glasgow, cause = 1)
plotAUCcurve(mskcc_pre.cdroc.glasgow, conf.int = FALSE, add = FALSE, col = pal["mskcc.pre"])
plotAUCcurve(mskcc_post.cdroc.glasgow, conf.int = FALSE, add = TRUE, col = pal["mskcc.post"])
plotAUCcurve(gg.cdroc.glasgow, conf.int = FALSE, add = TRUE, col = pal["gg"])
legend("topright", legend = c("Glasgow Preop", "Glasgow Postop", "GG"), col = c(pal["mskcc.pre"], pal["mskcc.post"], pal["gg"]))

```

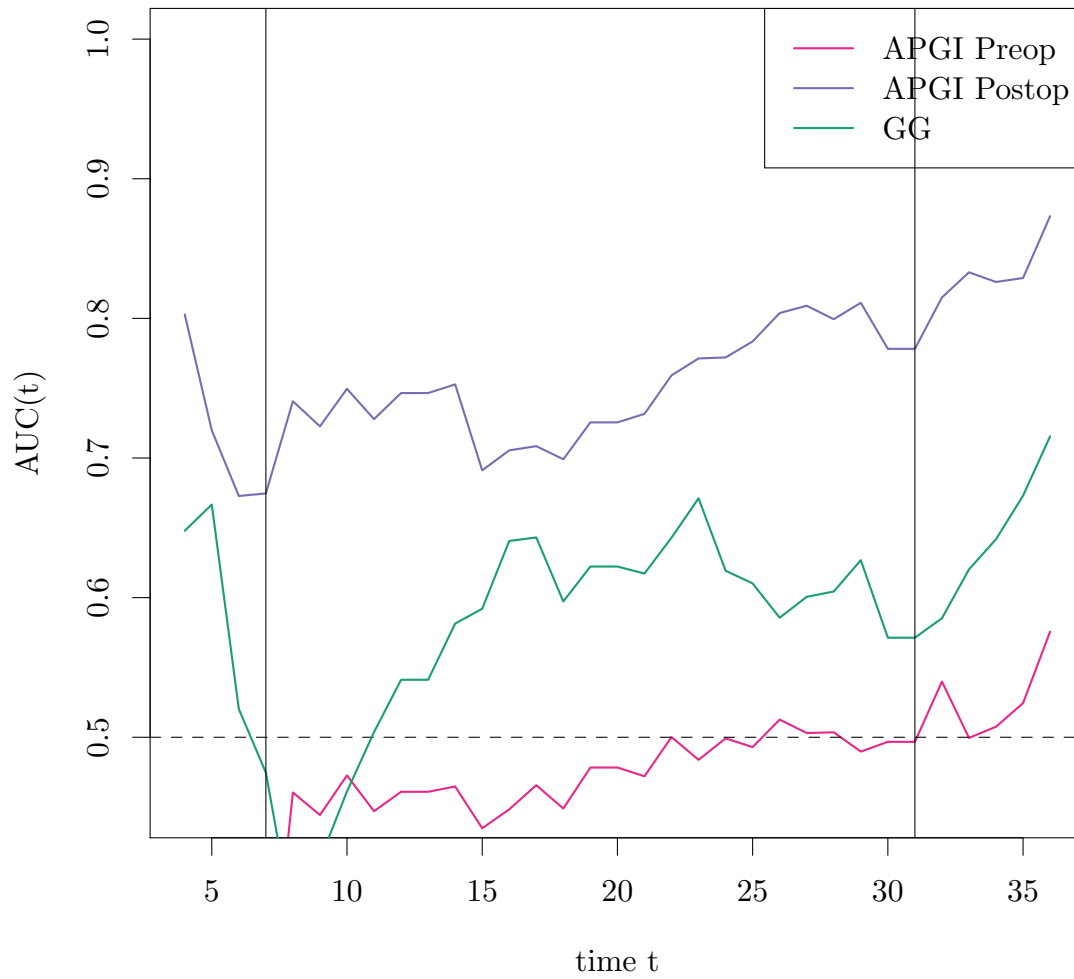




```

mskcc_pre.cdroc.apgi = timeROC(data.apgi$Time/365.25*12, data.apgi$DSD, mskcc_pre.linpred.apgi, cause =
mskcc_post.cdroc.apgi = timeROC(data.apgi$Time/365.25*12, data.apgi$DSD, mskcc_post.linpred.apgi, cause =
gg.cdroc.apgi = timeROC(data.apgi$Time/365.25*12, data.apgi$DSD, gg.linpred.apgi, cause = 1, times = sec
plotAUCcurve(mskcc_pre.cdroc.apgi, conf.int = FALSE, add = FALSE, col = pal["mskcc.pre"])
plotAUCcurve(mskcc_post.cdroc.apgi, conf.int = FALSE, add = TRUE, col = pal["mskcc.post"])
plotAUCcurve(gg.cdroc.apgi, conf.int = FALSE, add = TRUE, col = pal["gg"])
legend("topright", legend = c("APGI Preop", "APGI Postop", "GG"), col = c(pal["mskcc.pre"], pal["mskcc.p
abline(v = c(7, 31))

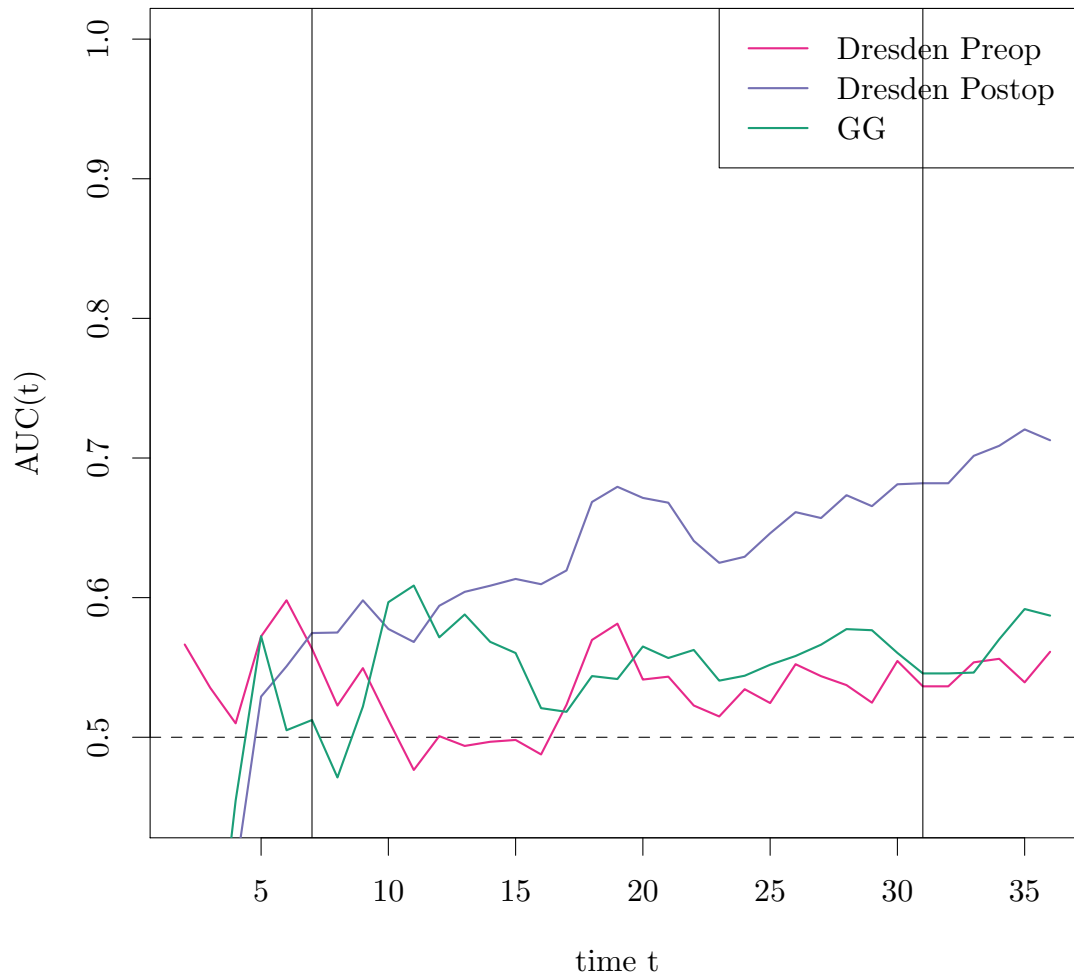
```



```

mskcc_pre.cdroc.dresden = timeROC(data.dresden$Time/365.25*12, data.dresden$DSD, mskcc_pre.linpred.dresden)
mskcc_post.cdroc.dresden = timeROC(data.dresden$Time/365.25*12, data.dresden$DSD, mskcc_post.linpred.dresden)
gg.cdroc.dresden = timeROC(data.dresden$Time/365.25*12, data.dresden$DSD, gg.linpred.dresden, cause = 1)
plotAUCcurve(mskcc_pre.cdroc.dresden, conf.int = FALSE, add = FALSE, col = pal["mskcc.pre"])
plotAUCcurve(mskcc_post.cdroc.dresden, conf.int = FALSE, add = TRUE, col = pal["mskcc.post"])
plotAUCcurve(gg.cdroc.dresden, conf.int = FALSE, add = TRUE, col = pal["gg"])
legend("topright", legend = c("Dresden Preop", "Dresden Postop", "GG"), col = c(pal["mskcc.pre"], pal["mskcc.post"], pal["gg"]))
abline(v = c(7, 31))

```



Incident-dynamic:

```
risksetROC.boot = function(time, event, marker, tmin = 0, tmax, B = 2000, ...)
{
  data = data.frame(time = time, event = event, marker = marker)
  eval_times = seq(tmin, tmax, length.out = 200)

  boot_obj = boot(data, function(data, indices) {
    data_draw = data[indices,]
    rsAUC = risksetAUC(Stime = data_draw$time, status = data_draw$event, marker = data_draw$marker)
    AUC_at_eval_times = approx(rsAUC$utimes, rsAUC$AUC, eval_times)$y
    AUC_at_eval_times
  }, R = B)

  res = list(boot = boot_obj, eval_times = eval_times)
  class(res) = "rrROC_boot"
  return(res)
}
```

```
plot.rrROC_boot = function(obj, add = FALSE, ci = FALSE, ci_conf = 0.95, ci_type = c("perc", "norm", "ba"))
{
  ci_type = match.arg(ci_type)
  boot_ci = t(apply(1:length(obj$boot$t0), function(i) {
```

```

ci = try(boot.ci(obj$boot, index = c(i, i), type = ci_type, conf = ci_conf))
if (class(ci) == "try-error") {
  if (ci_type == "norm") { return(c(NA, NA, NA)) } else { return(c(NA, NA, NA, NA, NA)) }
}
return(ci[[c("bca" = "bca", "norm" = "normal", "basic" = "basic", "stud" = "student", "p"]
}))
if (ci_type == "norm") { colnames(boot_ci) = c("level", "lci", "uci") } else { colnames(boot_ci) = c("level", "lci", "uci", "bci", "bci.lower", "bci.upper", "stud", "stud.lower", "stud.upper", "p"]
summ = as.data.frame(cbind(time = obj$eval_times, mean = obj$boot$t0, boot_ci))

if (!add) {
  plot(mean ~ time, summ, ylim = c(0.4, 1), type = "l", ...)
} else {
  lines(mean ~ time, summ, ...)
}
if (ci) {
  lines(lci ~ time, summ, lty = "dotted", ...)
  lines(uci ~ time, summ, lty = "dotted", ...)
}
abline(h = 0.5)
}

```

```

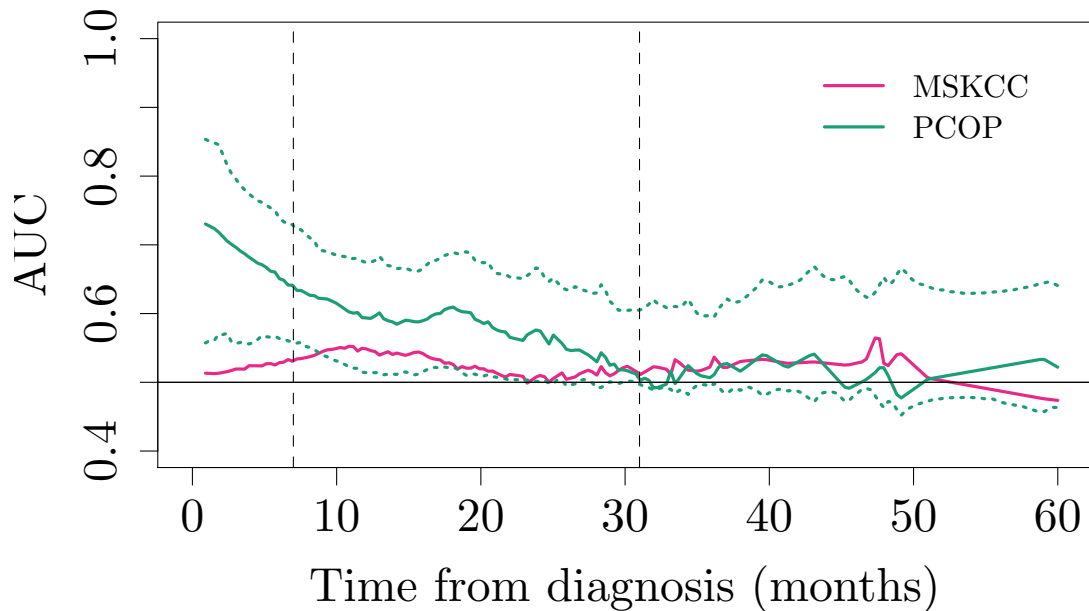
set.seed(20150216)
rrROC_boot.mskcc_pre.glasgow = risksetROC.boot(time = data.glasgow$Time/365.25*12, event = data.glasgow$DSD, mskcc_pre = TRUE)
rrROC_boot.mskcc_post.glasgow = risksetROC.boot(time = data.glasgow$Time/365.25*12, event = data.glasgow$DSD, mskcc_post = TRUE)
rrROC_boot.gg.glasgow = risksetROC.boot(time = data.glasgow$Time/365.25*12, event = data.glasgow$DSD, mskcc_pre = FALSE, mskcc_post = FALSE)

```

```

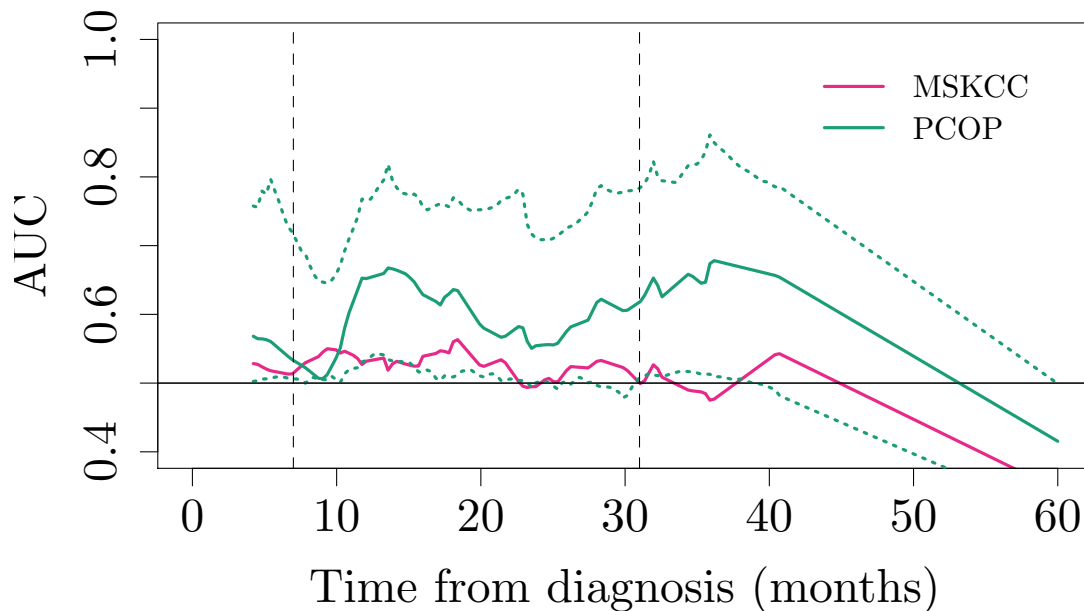
plot(rrROC_boot.mskcc_pre.glasgow, col = pal["mskcc.pre"], xlab = "Time from diagnosis (months)", ylab = "AUC", lwd = 3)
plot(rrROC_boot.gg.glasgow, col = pal["gg"], add = TRUE, ci = TRUE, lwd = 3)
legend("topright", legend = c("MSKCC", "PCOP"), col = c(pal["mskcc.pre"], pal["gg"]), lty = "solid", lwd = 3)
abline(v = c(7, 31), lty = "dashed")

```



```
set.seed(20150216)
rrROC_boot.mskcc_pre.apgi = risksetROC.boot(time = data.apgi$Time/365.25*12, event = data.apgi$DSD, marker = "MSKCC")
rrROC_boot.mskcc_post.apgi = risksetROC.boot(time = data.apgi$Time/365.25*12, event = data.apgi$DSD, marker = "PCOP")
rrROC_boot.gg.apgi = risksetROC.boot(time = data.apgi$Time/365.25*12, event = data.apgi$DSD, marker = "gg")
```

```
plot(rrROC_boot.mskcc_pre.apgi, col = pal["mskcc.pre"], xlab = "Time from diagnosis (months)", ylab = "AUC")
plot(rrROC_boot.gg.apgi, col = pal["gg"], add = TRUE, ci = TRUE, lwd = 3)
legend("topright", legend = c("MSKCC", "PCOP"), col = c(pal["mskcc.pre"], pal["gg"]), lty = "solid", lwd = 3)
abline(v = c(7, 31), lty = "dashed")
```



```
set.seed(20150216)
rrROC_boot.mskcc_pre.dresden = risksetROC.boot(time = data.dresden$Time/365.25*12, event = data.dresden$DSD, marker = "MSKCC")
rrROC_boot.mskcc_post.dresden = risksetROC.boot(time = data.dresden$Time/365.25*12, event = data.dresden$DSD, marker = "PCOP")
rrROC_boot.gg.dresden = risksetROC.boot(time = data.dresden$Time/365.25*12, event = data.dresden$DSD, marker = "gg")
```

```
plot(rrROC_boot.mskcc_pre.dresden, col = pal["mskcc.pre"], xlab = "Time from diagnosis (months)", ylab = "AUC")
plot(rrROC_boot.gg.dresden, col = pal["gg"], add = TRUE, ci = TRUE, lwd = 3)
legend("topright", legend = c("MSKCC", "PCOP"), col = c(pal["mskcc.pre"], pal["gg"]), lty = "solid", lwd = 3)
abline(v = c(7, 31), lty = "dashed")
```

