Report

P76074402

|  |  |  |  |
| --- | --- | --- | --- |
| IBM Quest Synthetic Data Generator(data.ntrans\_0.1.nitems\_0.1)  (min\_support ≥ 5%, min\_confidence ≥ 60%) | | | |
| My fpgrowth | Fpgrowth.exe | My fpgrowth | Fpgrowth.exe |
| {'38'} 43.43434 | 38 (43.4343) | {'81'} 21.21212 | 81 69 (8.08081) |
| {'63'} 32.32323 | 63 38 (16.1616) | {'38', '81'} 8.08081 | 81 38 (8.08081) |
| {'38', '63'} 16.16162 | 63 (32.3232) | {'69', '81'} 8.08081 | 81 36 (7.07071) |
| {'87'} 27.27273 | 87 38 (12.1212) | {'36', '81'} 7.07071 | 81 87 (7.07071) |
| {'87', '38'} 12.12121 | 87 63 (9.09091) | {'87', '81'} 7.07071 | 81 (21.2121) |
| {'87', '63'} 9.09091 | 87 (27.2727) | {'8'} 21.21212 | 8 38 (11.1111) |
| {'36'} 27.27273 | 36 38 (15.1515) | {'8', '38'} 11.11111 | 8 85 (8.08081) |
| {'36', '38'} 15.15152 | 36 63 38 (7.07071) | {'8', '63'} 8.08081 | 8 63 38 (6.06061) |
| {'36', '63'} 12.12121 | 36 63 (12.1212) | {'8', '38', '63'} 6.06061 | 8 63 (8.08081) |
| {'36', '38', '63'} 7.07071 | 36 87 38 (6.06061) | {'36', '8'} 5.05051 | 8 69 (6.06061) |
| {'87', '36'} 8.08081 | 36 87 (8.08081) | {'36', '38', '8'} 5.05051 | 8 87 (6.06061) |
| {'87', '36', '38'} 6.06061 | 36 (27.2727) | {'69', '8'} 6.06061 | 8 48 (5.05051) |
| {'69'} 26.26263 | 69 38 (11.1111) | {'8', '85'} 8.08081 | 8 36 38 (5.05051) |
| {'69', '38'} 11.11111 | 69 87 (9.09091) | {'8', '48'} 5.05051 | 8 36 (5.05051) |
| {'69', '63'} 8.08081 | 69 63 (8.08081) | {'87', '8'} 6.06061 | 8 (21.2121) |
| {'36', '69'} 7.07071 | 69 36 (7.07071) | {'17'} 20.20202 | 17 63 (9.09091) |
| {'87', '69'} 9.09091 | 69 (26.2626) | {'38', '17'} 7.07071 | 17 36 63 (6.06061) |
| {'48'} 24.24242 | 48 38 (16.1616) | {'17', '63'} 9.09091 | 17 36 (8.08081) |
| {'38', '48'} 16.16162 | 48 69 38 (5.05051) | {'36', '17'} 8.08081 | 17 8 (7.07071) |
| {'48', '63'} 8.08081 | 48 69 (8.08081) | {'36', '17', '63'} 6.06061 | 17 38 (7.07071) |
| {'38', '48', '63'} 6.06061 | 48 36 38 (5.05051) | {'8', '17'} 7.07071 | 17 48 36 (5.05051) |
| {'36', '48'} 8.08081 | 48 36 (8.08081) | {'85', '17'} 6.06061 | 17 48 (7.07071) |
| {'36', '38', '48'} 5.05051 | 48 63 38 (6.06061) | {'17', '48'} 7.07071 | 17 85 (6.06061) |
| {'36', '48', '63'} 5.05051 | 48 63 36 (5.05051) | {'36', '17', '48'} 5.05051 | 17 (20.202) |
| {'69', '48'} 8.08081 | 48 63 (8.08081) | {'11'} 20.20202 | 11 38 (8.08081) |
| {'69', '38', '48'} 5.05051 | 48 87 38 (5.05051) | {'11', '38'} 8.08081 | 11 63 (8.08081) |
| {'87', '48'} 6.06061 | 48 87 (6.06061) | {'11', '63'} 8.08081 | 11 81 (7.07071) |
| {'87', '38', '48'} 5.05051 | 48 (24.2424) | {'11', '81'} 7.07071 | 11 17 (6.06061) |
| {'85'} 22.22222 | 85 87 (9.09091) | {'11', '36'} 6.06061 | 11 36 (6.06061) |
| {'38', '85'} 8.08081 | 85 38 (8.08081) | {'11', '48'} 5.05051 | 11 48 (5.05051) |
| {'85', '63'} 6.06061 | 85 69 38 (5.05051) | {'11', '17'} 6.06061 | 11 (20.202) |
| {'36', '85'} 5.05051 | 85 69 (7.07071) | {'43'} 19.19192 | 43 63 (11.1111) |
| {'69', '85'} 7.07071 | 85 63 (6.06061) | {'38', '43'} 10.10101 | 43 38 63 (8.08081) |
| {'69', '38', '85'} 5.05051 | 85 36 (5.05051) | {'43', '63'} 11.11111 | 43 38 (10.101) |
| {'87', '85'} 9.09091 | 85 (22.2222) | {'38', '43', '63'} 8.08081 | 43 48 63 (7.07071) |
| {'36', '43'} 9.09091 | 43 48 38 63 (6.06061) | {'35', '63'} 7.07071 | 3 63 (5.05051) |
| {'36', '38', '43'} 7.07071 | 43 48 38 (7.07071) | {'35', '38', '63'} 7.07071 | 3 (17.1717) |
| {'36', '43', '63'} 6.06061 | 43 48 (9.09091) | {'69', '35'} 6.06061 | 80 38 (9.09091) |
| {'36', '38', '43', '63'} 5.05051 | 43 36 63 (6.06061) | {'36', '35'} 7.07071 | 80 3 (7.07071) |
| {'8', '43'} 7.07071 | 43 36 38 63 (5.05051) | {'35', '85'} 6.06061 | 80 48 (6.06061) |
| {'8', '38', '43'} 5.05051 | 43 36 38 (7.07071) | {'35', '48'} 5.05051 | 80 11 (5.05051) |
| {'17', '43'} 6.06061 | 43 36 48 (5.05051) | {'87', '35'} 5.05051 | 80 87 (5.05051) |
| {'17', '43', '63'} 5.05051 | 43 36 (9.09091) | {'3'} 17.17172 | 80 (17.1717) |
| {'69', '43'} 5.05051 | 43 8 38 (5.05051) | {'38', '3'} 8.08081 | 35 38 (10.101) |
| {'43', '48'} 9.09091 | 43 8 (7.07071) | {'3', '63'} 5.05051 | 35 36 (7.07071) |
| {'38', '43', '48'} 7.07071 | 43 17 63 (5.05051) | {'36', '3'} 5.05051 | 35 63 38 (7.07071) |
| {'43', '48', '63'} 7.07071 | 43 17 (6.06061) | {'8', '3'} 5.05051 | 35 63 (7.07071) |
| {'38', '43', '48', '63'} 6.06061 | 43 87 (6.06061) | {'17', '3'} 6.06061 | 35 85 (6.06061) |
| {'36', '43', '48'} 5.05051 | 43 69 (5.05051) | {'3', '48'} 5.05051 | 35 69 (6.06061) |
| {'87', '43'} 6.06061 | 43 (19.1919) | {'11', '3'} 6.06061 | 35 48 (5.05051) |
| {'28'} 19.19192 | 28 38 (7.07071) | {'43', '3'} 5.05051 | 35 87 (5.05051) |
| {'28', '38'} 7.07071 | 28 36 (7.07071) | {'80', '3'} 7.07071 | 35 (17.1717) |
| {'87', '28'} 6.06061 | 28 87 (6.06061) | {'72'} 16.16162 | 61 3 (8.08081) |
| {'28', '36'} 7.07071 | 28 48 (5.05051) | {'38', '72'} 5.05051 | 61 38 (7.07071) |
| {'28', '48'} 5.05051 | 28 69 (5.05051) | {'87', '72'} 7.07071 | 61 80 3 (5.05051) |
| {'28', '69'} 5.05051 | 28 (19.1919) | {'8', '72'} 5.05051 | 61 80 (7.07071) |
| {'83'} 18.18182 | 83 38 (7.07071) | {'69', '72'} 5.05051 | 61 63 (7.07071) |
| {'38', '83'} 7.07071 | 83 63 (7.07071) | {'61'} 16.16162 | 61 43 (6.06061) |
| {'83', '63'} 7.07071 | 83 17 (6.06061) | {'38', '61'} 7.07071 | 61 48 (6.06061) |
| {'69', '83'} 5.05051 | 83 48 (6.06061) | {'61', '63'} 7.07071 | 61 36 63 (5.05051) |
| {'36', '83'} 5.05051 | 83 81 (5.05051) | {'36', '61'} 6.06061 | 61 36 (6.06061) |
| {'17', '83'} 6.06061 | 83 85 (5.05051) | {'61', '36', '63'} 5.05051 | 61 35 (5.05051) |
| {'85', '83'} 5.05051 | 83 69 (5.05051) | {'35', '61'} 5.05051 | 61 (16.1616) |
| {'83', '48'} 6.06061 | 83 36 (5.05051) | {'3', '61'} 8.08081 | 40 38 (8.08081) |
| {'81', '83'} 5.05051 | 83 (18.1818) | {'61', '48'} 6.06061 | 40 43 (5.05051) |
| {'80'} 17.17172 | 3 38 (8.08081) | {'43', '61'} 6.06061 | 40 8 (5.05051) |
| {'38', '80'} 9.09091 | 3 11 (6.06061) | {'80', '61'} 7.07071 | 40 85 (5.05051) |
| {'80', '48'} 6.06061 | 3 17 (6.06061) | {'80', '3', '61'} 5.05051 | 40 48 (5.05051) |
| {'11', '80'} 5.05051 | 3 43 (5.05051) | {'40'} 16.16162 | 40 (16.1616) |
| {'87', '80'} 5.05051 | 3 8 (5.05051) | {'40', '38'} 8.08081 | 72 87 (7.07071) |
| {'35'} 17.17172 | 3 48 (5.05051) | {'40', '48'} 5.05051 | 72 8 (5.05051) |
| {'35', '38'} 10.10101 | 3 36 (5.05051) | {'40', '85'} 5.05051 | 72 69 (5.05051) |
| {'40', '8'} 5.05051 | 72 38 (5.05051) | {'62', '63'} 9.09091 | 62 38 (7.07071) |
| {'40', '43'} 5.05051 | 72 (16.1616) | {'62', '38', '63'} 5.05051 | 62 36 (5.05051) |
| {'93'} 15.15152 | 89 38 (6.06061) | {'62', '36'} 5.05051 | 62 87 63 (5.05051) |
| {'38', '93'} 7.07071 | 89 87 (6.06061) | {'87', '62'} 5.05051 | 62 87 (5.05051) |
| {'69', '93'} 7.07071 | 89 72 (5.05051) | {'87', '62', '63'} 5.05051 | 62 (14.1414) |
| {'93', '85'} 5.05051 | 89 80 (5.05051) | {'14'} 14.14141 | 14 38 (10.101) |
| {'93', '48'} 5.05051 | 89 28 (5.05051) | {'38', '14'} 10.10101 | 14 36 38 (5.05051) |
| {'89'} 15.15152 | 89 85 (5.05051) | {'36', '14'} 7.07071 | 14 36 (7.07071) |
| {'38', '89'} 6.06061 | 89 48 (5.05051) | {'36', '38', '14'} 5.05051 | 14 11 (5.05051) |
| {'89', '80'} 5.05051 | 89 (15.1515) | {'11', '14'} 5.05051 | 14 (14.1414) |
| {'89', '48'} 5.05051 | 39 87 (8.08081) | {'12'} 14.14141 | 12 38 (8.08081) |
| {'93', '89'} 5.05051 | 39 28 (6.06061) | {'38', '12'} 8.08081 | 12 63 (6.06061) |
| {'89', '72'} 5.05051 | 39 72 (5.05051) | {'12', '63'} 6.06061 | 12 85 (5.05051) |
| {'87', '89'} 6.06061 | 39 8 (5.05051) | {'85', '12'} 5.05051 | 12 48 (5.05051) |
| {'28', '89'} 5.05051 | 39 85 (5.05051) | {'12', '48'} 5.05051 | 12 87 (5.05051) |
| {'89', '85'} 5.05051 | 39 69 (5.05051) | {'87', '12'} 5.05051 | 12 (14.1414) |
| {'74'} 15.15152 | 39 (15.1515) | {'95'} 13.13131 | 23 11 (5.05051) |
| {'38', '74'} 7.07071 | 74 38 (7.07071) | {'95', '3'} 6.06061 | 23 69 (5.05051) |
| {'74', '63'} 5.05051 | 74 48 (6.06061) | {'85', '95'} 6.06061 | 23 36 (5.05051) |
| {'69', '74'} 5.05051 | 74 40 (5.05051) | {'72', '95'} 6.06061 | 23 38 (5.05051) |
| {'36', '74'} 5.05051 | 74 69 (5.05051) | {'71'} 13.13131 | 23 (13.1313) |
| {'74', '48'} 6.06061 | 74 36 (5.05051) | {'71', '38'} 7.07071 | 25 38 (8.08081) |
| {'40', '74'} 5.05051 | 74 63 (5.05051) | {'87', '71'} 7.07071 | 25 72 (5.05051) |
| {'39'} 15.15152 | 74 (15.1515) | {'36', '71'} 5.05051 | 25 36 (5.05051) |
| {'8', '39'} 5.05051 | 93 38 (7.07071) | {'71', '80'} 5.05051 | 25 87 (5.05051) |
| {'28', '39'} 6.06061 | 93 69 (7.07071) | {'51'} 13.13131 | 25 63 (5.05051) |
| {'87', '39'} 8.08081 | 93 89 (5.05051) | {'38', '51'} 5.05051 | 25 (13.1313) |
| {'39', '72'} 5.05051 | 93 85 (5.05051) | {'51', '63'} 6.06061 | 95 85 (6.06061) |
| {'69', '39'} 5.05051 | 93 48 (5.05051) | {'36', '51'} 5.05051 | 95 72 (6.06061) |
| {'39', '85'} 5.05051 | 93 (15.1515) | {'87', '51'} 5.05051 | 95 3 (6.06061) |
| {'86'} 14.14141 | 86 35 (5.05051) | {'47'} 13.13131 | 95 23 (5.05051) |
| {'28', '86'} 5.05051 | 86 28 (5.05051) | {'38', '47'} 7.07071 | 95 (13.1313) |
| {'35', '86'} 5.05051 | 86 48 (5.05051) | {'47', '80'} 6.06061 | 47 38 (7.07071) |
| {'87', '86'} 5.05051 | 86 87 (5.05051) | {'47', '3'} 5.05051 | 47 80 (6.06061) |
| {'86', '48'} 5.05051 | 86 (14.1414) | {'25'} 13.13131 | 47 3 (5.05051) |
| {'62'} 14.14141 | 62 63 (9.09091) | {'38', '25'} 8.08081 | 47 (13.1313) |
| {'62', '38'} 7.07071 | 62 38 63 (5.05051) | {'25', '63'} 5.05051 | 71 38 (7.07071) |
| {'36', '25'} 5.05051 | 71 87 (7.07071) | {'68'} 9.09091 | 66 85 (5.05051) |
| {'51', '25'} 5.05051 | 71 80 (5.05051) | {'38', '68'} 6.06061 | 66 (9.09091) |
| {'87', '25'} 5.05051 | 71 36 (5.05051) | {'36', '68'} 5.05051 | 42 (9.09091) |
| {'72', '25'} 5.05051 | 71 (13.1313) | {'66'} 9.09091 | 0 11 (6.06061) |
| {'23'} 13.13131 | 51 63 (6.06061) | {'85', '66'} 5.05051 | 0 (9.09091) |
| {'38', '23'} 5.05051 | 51 25 (5.05051) | {'42'} 9.09091 | 68 38 (6.06061) |
| {'36', '23'} 5.05051 | 51 36 (5.05051) | {'0'} 9.09091 | 68 36 (5.05051) |
| {'11', '23'} 5.05051 | 51 87 (5.05051) | {'11', '0'} 6.06061 | 68 (9.09091) |
| {'23', '95'} 5.05051 | 51 38 (5.05051) | {'97'} 8.08081 | 67 (8.08081) |
| {'69', '23'} 5.05051 | 51 (13.1313) | {'38', '97'} 5.05051 | 97 38 (5.05051) |
| {'78'} 12.12121 | 78 36 (7.07071) | {'73'} 8.08081 | 97 (8.08081) |
| {'38', '78'} 7.07071 | 78 38 (7.07071) | {'67'} 8.08081 | 26 63 (5.05051) |
| {'36', '78'} 7.07071 | 78 (12.1212) | {'26'} 8.08081 | 26 (8.08081) |
| {'7'} 12.12121 | 29 11 (5.05051) | {'26', '63'} 5.05051 | 73 (8.08081) |
| {'7', '69'} 5.05051 | 29 (12.1212) | {'91'} 7.07071 | 91 (7.07071) |
| {'7', '85'} 5.05051 | 7 85 (5.05051) | {'50'} 7.07071 | 50 (7.07071) |
| {'29'} 12.12121 | 7 69 (5.05051) | {'45'} 7.07071 | 45 (7.07071) |
| {'11', '29'} 5.05051 | 7 (12.1212) | {'4'} 7.07071 | 4 38 (5.05051) |
| {'21'} 12.12121 | 21 80 (6.06061) | {'38', '4'} 5.05051 | 4 (7.07071) |
| {'21', '63'} 5.05051 | 21 63 (5.05051) | {'90'} 6.06061 | 34 (6.06061) |
| {'21', '80'} 6.06061 | 21 (12.1212) | {'34'} 6.06061 | 90 (6.06061) |
| {'9'} 11.11111 | 57 85 (5.05051) | {'31'} 6.06061 | 27 (6.06061) |
| {'9', '38'} 5.05051 | 57 (11.1111) | {'27'} 6.06061 | 18 (6.06061) |
| {'36', '9'} 6.06061 | 9 36 (6.06061) | {'18'} 6.06061 | 31 (6.06061) |
| {'9', '48'} 5.05051 | 9 17 (5.05051) | {'96'} 5.05051 | 70 (5.05051) |
| {'9', '17'} 5.05051 | 9 48 (5.05051) | {'92'} 5.05051 | 6 (5.05051) |
| {'57'} 11.11111 | 9 38 (5.05051) | {'70'} 5.05051 | 96 (5.05051) |
| {'57', '85'} 5.05051 | 9 (11.1111) | {'6'} 5.05051 | 20 (5.05051) |
| {'5'} 11.11111 | 13 (11.1111) | {'55'} 5.05051 | 92 (5.05051) |
| {'40', '5'} 5.05051 | 5 40 (5.05051) | {'49'} 5.05051 | 46 (5.05051) |
| {'87', '5'} 5.05051 | 5 87 (5.05051) | {'46'} 5.05051 | 49 (5.05051) |
| {'33'} 11.11111 | 5 (11.1111) | {'20'} 5.05051 | 55 (5.05051) |
| {'33', '63'} 6.06061 | 33 87 (7.07071) |
| {'87', '33'} 7.07071 | 33 63 (6.06061) |
| {'13'} 11.11111 | 33 (11.1111) |
| {'52'} 10.10101 | 52 36 (6.06061) |
| {'36', '52'} 6.06061 | 52 (10.101) |

**fpgrowth -s5 -ts -c50 -f, output0101.csv output\_01\_01.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm**

**version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt**

**reading output0101.csv ... [93 item(s), 99 transaction(s)] done [0.00s].**

**filtering, sorting and recoding items ... [69 item(s)] done [0.00s].**

**sorting and reducing transactions ... [99 transaction(s)] done [0.00s].**

**writing output\_01\_01.out ... [287 set(s)] done [0.00s].**

|  |  |  |  |
| --- | --- | --- | --- |
| Association rules  (min\_support ≥ 5%, min\_confidence ≥ 60%) | | | |
| 38 <- 36 87(6.06061,75) | 48 <- 17 36(5.05051,62.5) | 48 <- 43 63 38(6.06061,75) | 61 <- 80 3(5.05051,71.4286) |
| 38 <- 48(16.1616,66.6667) | 17 <- 48 36(5.05051,62.5) | 43 <- 48 63 38(6.06061,100) | 63 <- 62 38(5.05051,71.4286) |
| 38 <- 48 63(6.06061,75) | 38 <- 43 63(8.08081,72.7273) | 63 <- 43 48(7.07071,77.7778) | 63 <- 62(9.09091,64.2857) |
| 38 <- 48 87(5.05051,83.3333) | 63 <- 43 38(8.08081,80) | 48 <- 43 63(7.07071,63.6364) | 63 <- 62 87(5.05051,100) |
| 38 <- 48 36(5.05051,62.5) | 38 <- 43 36(7.07071,77.7778) | 43 <- 48 63(7.07071,87.5) | 38 <- 14(10.101,71.4286) |
| 63 <- 48 36(5.05051,62.5) | 36 <- 43 38(7.07071,70) | 43 <- 48 36(5.05051,62.5) | 38 <- 14 36(5.05051,71.4286) |
| 36 <- 48 63(5.05051,62.5) | 38 <- 43 36 63(5.05051,83.3333) | 38 <- 43 8(5.05051,71.4286) | 38 <- 25(8.08081,61.5385) |
| 38 <- 48 69(5.05051,62.5) | 63 <- 43 36 38(5.05051,71.4286) | 63 <- 43 17(5.05051,83.3333) | 87 <- 33(7.07071,63.6364) |
| 38 <- 85 69(5.05051,71.4286) | 36 <- 43 63 38(5.05051,62.5) | 38 <- 35 63(7.07071,100) | 36 <- 52(6.06061,60) |
| 69 <- 85 38(5.05051,62.5) | 43 <- 36 63 38(5.05051,71.4286) | 63 <- 35 38(7.07071,70) | 11 <- 0(6.06061,66.6667) |
| 38 <- 8 63(6.06061,75) | 63 <- 43 36(6.06061,66.6667) | 63 <- 61 36(5.05051,83.3333) | 38 <- 68(6.06061,66.6667) |
| 38 <- 8 36(5.05051,100) | 38 <- 43 48(7.07071,77.7778) | 36 <- 61 63(5.05051,71.4286) | 38 <- 97(5.05051,62.5) |
| 63 <- 17 36(6.06061,75) | 48 <- 43 38(7.07071,70) | 3 <- 61 80(5.05051,71.4286) | 63 <- 26(5.05051,62.5) |
| 36 <- 17 63(6.06061,66.6667) | 38 <- 43 48 63(6.06061,85.7143) | 80 <- 61 3(5.05051,62.5) | 38 <- 4(5.05051,71.4286) |
| 36 <- 17 48(5.05051,71.4286) | 63 <- 43 48 38(6.06061,85.7143) |  |  |

**fpgrowth -s5 -tr -v(%S,%C) -o -c60 -f, output0101.csv output\_rule\_0101.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm**

**version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt**

**reading output0101.csv ... [93 item(s), 99 transaction(s)] done [0.00s].**

**filtering, sorting and recoding items ... [69 item(s)] done [0.00s].**

**sorting and reducing transactions ... [99 transaction(s)] done [0.00s].**

**finding frequent item set(s) ... done [0.00s].**

**writing output\_rule\_0101.out ... [58 rule(s)] done [0.00s].**

|  |  |  |  |
| --- | --- | --- | --- |
| IBM Quest Synthetic Data Generator(data.ntrans\_0.1.nitems\_1)  (min\_support ≥ 5%, min\_confidence ≥ 60%) | | | |
| My fpgrowth | Fpgrowth.exe | My fpgrowth | Fpgrowth.exe |
| {'124'} 9.09091 | 124 (9.09091) | {'790'} 5.05051 | 127 (5.05051) |
| {'416'} 8.08081 | 416 (8.08081) | {'772'} 5.05051 | 994 (5.05051) |
| {'806'} 7.07071 | 806 (7.07071) | {'673'} 5.05051 | 578 (5.05051) |
| {'778'} 7.07071 | 778 (7.07071) | {'60'} 5.05051 | 813 (5.05051) |
| {'682'} 7.07071 | 592 (7.07071) | {'578'} 5.05051 | 36 (5.05051) |
| {'592'} 7.07071 | 682 (7.07071) | {'38'} 5.05051 | 934 (5.05051) |
| {'988'} 6.06061 | 874 (6.06061) | {'36'} 5.05051 | 820 (5.05051) |
| {'874'} 6.06061 | 707 (6.06061) | {'316'} 5.05051 | 673 (5.05051) |
| {'707'} 6.06061 | 214 (6.06061) | {'306'} 5.05051 | 38 (5.05051) |
| {'456'} 6.06061 | 988 (6.06061) | {'238'} 5.05051 | 772 (5.05051) |
| {'214'} 6.06061 | 456 (6.06061) | {'132'} 5.05051 | 790 (5.05051) |
| {'994'} 5.05051 | 238 (5.05051) | {'127'} 5.05051 | 60 (5.05051) |
| {'981'} 5.05051 | 132 (5.05051) | {'123'} 5.05051 | 837 (5.05051) |
| {'934'} 5.05051 | 306 (5.05051) | {'119'} 5.05051 | 858 (5.05051) |
| {'858'} 5.05051 | 117 (5.05051) | {'117'} 5.05051 | 123 (5.05051) |
| {'837'} 5.05051 | 316 (5.05051) |  |  |
| {'820'} 5.05051 | 981 (5.05051) |  |  |
| {'813'} 5.05051 | 119 (5.05051) |  |  |

**fpgrowth -s5 -ts -c50 -f, output011.csv output\_01\_1.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm**

**version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt**

**reading output011.csv ... [499 item(s), 99 transaction(s)] done [0.00s].**

**filtering, sorting and recoding items ... [33 item(s)] done [0.00s].**

**sorting and reducing transactions ... [74/99 transaction(s)] done [0.00s].**

**writing output\_01\_1.out ... [33 set(s)] done [0.00s].**

**Association rules: 0 rule**

**fpgrowth -s5 -tr -v(%S,%C) -o -c60 -f, output011.csv output\_rule\_011.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm**

**version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt**

**reading output011.csv ... [499 item(s), 99 transaction(s)] done [0.00s].**

**filtering, sorting and recoding items ... [33 item(s)] done [0.00s].**

**sorting and reducing transactions ... [74/99 transaction(s)] done [0.00s].**

**finding frequent item set(s) ... done [0.00s].**

**writing output\_rule\_011.out ... [0 rule(s)] done [0.00s].**

|  |  |  |  |
| --- | --- | --- | --- |
| IBM Quest Synthetic Data Generator(data.ntrans\_1.nitems\_0.1)  (min\_support ≥ 5%, min\_confidence ≥ 60%) | | | |
| My fpgrowth | Fpgrowth.exe | My fpgrowth | Fpgrowth.exe |
| {'38'} 37.90650 | 38 (37.9065) | {'81', '69'} 5.79268 | 81 63 (5.69106) |
| {'63'} 36.38211 | 63 38 (14.2276) | {'81', '36'} 5.38618 | 81 36 (5.38618) |
| {'63', '38'} 14.22764 | 63 (36.3821) | {'87', '81'} 5.89431 | 81 (15.752) |
| {'87'} 35.36585 | 87 38 (14.7358) | {'40'} 15.75203 | 40 38 (5.58943) |
| {'87', '38'} 14.73577 | 87 63 38 (5.58943) | {'40', '38'} 5.58943 | 40 87 (5.0813) |
| {'63', '87'} 13.61789 | 87 63 (13.6179) | {'87', '40'} 5.08130 | 40 (15.752) |
| {'63', '87', '38'} 5.58943 | 87 (35.3659) | {'14'} 15.65041 | 14 38 (6.70732) |
| {'69'} 30.28455 | 69 63 (11.7886) | {'14', '38'} 6.70732 | 14 63 (6.09756) |
| {'38', '69'} 11.28049 | 69 38 (11.2805) | {'63', '14'} 6.09756 | 14 87 (5.28455) |
| {'63', '69'} 11.78862 | 69 87 (10.2642) | {'87', '14'} 5.28455 | 14 (15.6504) |
| {'87', '69'} 10.26423 | 69 (30.2846) | {'61'} 15.54878 | 61 38 (6.50407) |
| {'36'} 27.23577 | 36 38 (10.874) | {'38', '61'} 6.50407 | 61 63 (6.19919) |
| {'38', '36'} 10.87398 | 36 63 (10.874) | {'63', '61'} 6.19919 | 61 (15.5488) |
| {'63', '36'} 10.87398 | 36 87 (10.3659) | {'86'} 15.44715 | 86 38 (6.60569) |
| {'36', '69'} 9.34959 | 36 69 (9.34959) | {'86', '38'} 6.60569 | 86 87 (6.30081) |
| {'87', '36'} 10.36585 | 36 (27.2358) | {'86', '69'} 5.48780 | 86 69 (5.4878) |
| {'8'} 23.06911 | 8 87 (8.94309) | {'87', '86'} 6.30081 | 86 (15.4472) |
| {'8', '38'} 8.84146 | 8 38 (8.84146) | {'62'} 15.24390 | 62 87 (6.91057) |
| {'63', '8'} 8.43496 | 8 63 (8.43496) | {'38', '62'} 6.91057 | 62 38 (6.91057) |
| {'8', '69'} 6.80894 | 8 36 (7.11382) | {'63', '62'} 6.40244 | 62 63 (6.40244) |
| {'8', '36'} 7.11382 | 8 69 (6.80894) | {'87', '62'} 6.91057 | 62 (15.2439) |
| {'87', '8'} 8.94309 | 8 (23.0691) | {'67'} 14.93902 | 67 63 (6.50407) |
| {'48'} 22.96748 | 48 69 (9.04472) | {'67', '38'} 6.40244 | 67 38 (6.40244) |
| {'48', '38'} 8.53659 | 48 87 (8.73984) | {'63', '67'} 6.50407 | 67 69 (5.99593) |
| {'63', '48'} 8.33333 | 48 38 (8.53659) | {'67', '69'} 5.99593 | 67 87 (5.38618) |
| {'48', '69'} 9.04472 | 48 63 (8.33333) | {'87', '67'} 5.38618 | 67 (14.939) |
| {'48', '36'} 7.01220 | 48 36 (7.0122) | {'78'} 14.43089 | 29 38 (5.79268) |
| {'48', '8'} 6.19919 | 48 8 (6.19919) | {'78', '38'} 5.48780 | 29 87 (5.58943) |
| {'87', '48'} 8.73984 | 48 (22.9675) | {'63', '78'} 5.89431 | 29 69 (5.28455) |
| {'28'} 21.03659 | 28 38 (9.14634) | {'78', '36'} 5.18293 | 29 (14.4309) |
| {'28', '38'} 9.14634 | 28 63 (8.63821) | {'29'} 14.43089 | 78 63 (5.89431) |
| {'63', '28'} 8.63821 | 28 87 (7.72358) | {'38', '29'} 5.79268 | 78 38 (5.4878) |
| {'28', '69'} 6.60569 | 28 36 (7.62195) | {'29', '69'} 5.28455 | 78 36 (5.18293) |
| {'28', '36'} 7.62195 | 28 69 (6.60569) | {'87', '29'} 5.58943 | 78 (14.4309) |
| {'8', '28'} 5.58943 | 28 48 (5.58943) | {'23'} 14.02439 | 23 38 (5.89431) |
| {'48', '28'} 5.58943 | 28 8 (5.58943) | {'38', '23'} 5.89431 | 23 (14.0244) |
| {'87', '28'} 7.72358 | 28 (21.0366) | {'72'} 13.82114 | 51 63 (5.4878) |
| {'85'} 20.12195 | 85 38 (7.72358) | {'38', '72'} 5.28455 | 51 38 (5.38618) |
| {'85', '38'} 7.72358 | 85 69 (7.0122) | {'63', '72'} 5.58943 | 51 87 (5.28455) |
| {'63', '85'} 7.01220 | 85 63 (7.0122) | {'87', '72'} 5.79268 | 51 (13.8211) |
| {'85', '69'} 7.01220 | 85 87 (6.60569) | {'51'} 13.82114 | 72 87 (5.79268) |
| {'85', '36'} 5.69106 | 85 36 (5.69106) | {'38', '51'} 5.38618 | 72 63 (5.58943) |
| {'87', '85'} 6.60569 | 85 (20.122) | {'63', '51'} 5.48780 | 72 38 (5.28455) |
| {'3'} 19.91870 | 3 38 (7.92683) | {'87', '51'} 5.28455 | 72 (13.8211) |
| {'38', '3'} 7.92683 | 3 87 (7.62195) | {'71'} 13.51626 | 71 38 (5.79268) |
| {'63', '3'} 6.70732 | 3 63 (6.70732) | {'71', '38'} 5.79268 | 71 87 (5.69106) |
| {'3', '69'} 5.69106 | 3 36 (5.89431) | {'87', '71'} 5.69106 | 71 (13.5163) |
| {'36', '3'} 5.89431 | 3 69 (5.69106) | {'35'} 13.41463 | 35 63 (6.40244) |
| {'8', '3'} 5.18293 | 3 8 (5.18293) | {'38', '35'} 5.89431 | 35 38 (5.89431) |
| {'87', '3'} 7.62195 | 3 (19.9187) | {'63', '35'} 6.40244 | 35 87 (5.4878) |
| {'17'} 19.71545 | 17 38 (7.62195) | {'87', '35'} 5.48780 | 35 (13.4146) |
| {'17', '38'} 7.62195 | 17 87 (7.52033) | {'93'} 13.10976 | 93 38 (5.69106) |
| {'63', '17'} 6.91057 | 17 63 (6.91057) | {'93', '38'} 5.69106 | 93 87 (5.28455) |
| {'17', '69'} 5.18293 | 17 36 (6.30081) | {'87', '93'} 5.28455 | 93 (13.1098) |
| {'17', '36'} 6.30081 | 17 69 (5.18293) | {'89'} 13.00813 | 73 38 (5.58943) |
| {'17', '8'} 5.08130 | 17 8 (5.0813) | {'73'} 13.00813 | 73 87 (5.28455) |
| {'87', '17'} 7.52033 | 17 (19.7154) | {'73', '38'} 5.58943 | 73 (13.0081) |
| {'47'} 17.88618 | 11 38 (8.23171) | {'87', '73'} 5.28455 | 89 (13.0081) |
| {'47', '38'} 6.40244 | 11 63 (6.50407) | {'7'} 11.48374 | 7 38 (5.0813) |
| {'63', '47'} 7.11382 | 11 87 (5.58943) | {'38', '7'} 5.08130 | 7 (11.4837) |
| {'47', '36'} 5.79268 | 11 36 (5.38618) | {'52'} 11.07724 | 52 (11.0772) |
| {'87', '47'} 6.91057 | 11 (17.8862) | {'95'} 10.77236 | 95 (10.7724) |
| {'11'} 17.88618 | 47 63 (7.11382) | {'21'} 10.56911 | 21 (10.5691) |
| {'11', '38'} 8.23171 | 47 87 (6.91057) | {'97'} 9.55285 | 97 (9.55285) |
| {'63', '11'} 6.50407 | 47 38 (6.40244) | {'33'} 9.45122 | 33 (9.45122) |
| {'11', '36'} 5.38618 | 47 36 (5.79268) | {'66'} 9.34959 | 26 (9.34959) |
| {'87', '11'} 5.58943 | 47 (17.8862) | {'26'} 9.34959 | 66 (9.34959) |
| {'83'} 16.97154 | 83 63 (6.60569) | {'42'} 9.04472 | 42 (9.04472) |
| {'38', '83'} 6.19919 | 83 38 (6.19919) | {'74'} 8.94309 | 34 (8.94309) |
| {'63', '83'} 6.60569 | 83 87 (5.58943) | {'34'} 8.94309 | 74 (8.94309) |
| {'83', '69'} 5.48780 | 83 36 (5.4878) | {'12'} 8.73984 | 12 (8.73984) |
| {'36', '83'} 5.48780 | 83 69 (5.4878) | {'25'} 8.23171 | 25 (8.23171) |
| {'87', '83'} 5.58943 | 83 (16.9715) | {'68'} 8.02846 | 68 (8.02846) |
| {'80'} 16.86992 | 80 38 (6.70732) | {'70'} 7.92683 | 70 (7.92683) |
| {'38', '80'} 6.70732 | 80 87 (6.60569) | {'0'} 7.82520 | 0 (7.8252) |
| {'63', '80'} 5.99593 | 80 63 (5.99593) | {'5'} 7.72358 | 5 (7.72358) |
| {'80', '69'} 5.08130 | 80 69 (5.0813) | {'60'} 7.62195 | 60 (7.62195) |
| {'87', '80'} 6.60569 | 80 (16.8699) | {'18'} 7.41870 | 18 (7.4187) |
| {'43'} 16.46341 | 43 87 (6.91057) | {'49'} 7.31707 | 49 (7.31707) |
| {'43', '38'} 6.09756 | 43 63 (6.60569) | {'13'} 7.21545 | 13 (7.21545) |
| {'63', '43'} 6.60569 | 43 38 (6.09756) | {'6'} 6.91057 | 6 (6.91057) |
| {'43', '69'} 5.28455 | 43 69 (5.28455) | {'50'} 6.91057 | 50 (6.91057) |
| {'87', '43'} 6.91057 | 43 (16.4634) | {'10'} 6.70732 | 10 (6.70732) |
| {'39'} 15.95528 | 39 87 (7.0122) | {'57'} 6.50407 | 57 (6.50407) |
| {'38', '39'} 6.40244 | 39 38 (6.40244) | {'53'} 6.40244 | 53 (6.40244) |
| {'63', '39'} 5.99593 | 39 63 (5.99593) | {'90'} 6.30081 | 90 (6.30081) |
| {'87', '39'} 7.01220 | 39 (15.9553) | {'45'} 5.89431 | 45 (5.89431) |
| {'9'} 15.75203 | 9 87 (6.80894) | {'55'} 5.58943 | 24 (5.58943) |
| {'9', '38'} 5.69106 | 9 63 (6.40244) | {'24'} 5.58943 | 55 (5.58943) |
| {'63', '9'} 6.40244 | 9 38 (5.69106) | {'19'} 5.48780 | 19 (5.4878) |
| {'87', '9'} 6.80894 | 9 (15.752) | {'4'} 5.38618 | 4 (5.38618) |
| {'81'} 15.75203 | 81 38 (6.60569) | {'41'} 5.18293 | 41 (5.18293) |
| {'81', '38'} 6.60569 | 81 87 (5.89431) | {'31'} 5.08130 | 31 (5.0813) |
| {'63', '81'} 5.69106 | 81 69 (5.79268) |  |  |

**fpgrowth -s5 -ts -c50 -f, output101.csv output\_1\_01.out fpgrowth - find frequent item sets with the fpgrowth algorithm version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt reading output101.csv ... [99 item(s), 984 transaction(s)] done [0.00s]. filtering, sorting and recoding items ... [69 item(s)] done [0.00s]. sorting and reducing transactions ... [981/984 transaction(s)] done [0.00s]. writing output\_1\_01.out ... [187 set(s)] done [0.00s].**

**Association rules: 0 rule**

**fpgrowth -s5 -tr -v(%S,%C) -o -c60 -f, output101.csv output\_rule\_101.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm**

**version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt**

**reading output101.csv ... [99 item(s), 984 transaction(s)] done [0.00s].**

**filtering, sorting and recoding items ... [69 item(s)] done [0.00s].**

**sorting and reducing transactions ... [981/984 transaction(s)] done [0.00s].**

**finding frequent item set(s) ... done [0.00s].**

**writing output\_rule\_101.out ... [0 rule(s)] done [0.00s].**

|  |  |
| --- | --- |
| IBM Quest Synthetic Data Generator(data.ntrans\_1.nitems\_1)  (min\_support ≥ 5%, min\_confidence ≥ 60%) | |
| My fpgrowth | Fpgrowth.exe |
| {'592'} 7.06244 | 592 (7.06244) |
| {'709'} 6.65302 | 709 (6.65302) |
| {'553'} 6.14125 | 553 (6.14125) |
| {'416'} 5.73183 | 416 (5.73183) |
| {'127'} 5.52712 | 127 (5.52712) |
| {'571'} 5.22006 | 571 (5.22006) |
| {'209'} 5.22006 | 209 (5.22006) |
| {'707'} 5.11771 | 707 (5.11771) |
| {'238'} 5.01535 | 238 (5.01535) |

**fpgrowth -s5 -ts -c50 -f, output11.csv output\_1\_1.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt reading output11.csv ... [891 item(s), 977 transaction(s)] done [0.00s]. filtering, sorting and recoding items ... [9 item(s)] done [0.00s]. sorting and reducing transactions ... [50/977 transaction(s)] done [0.00s]. writing output\_1\_1.out ... [9 set(s)] done [0.00s].**

**Association rules: 0 rule**

**fpgrowth -s5 -tr -v(%S,%C) -o -c60 -f, output11.csv output\_rule\_11.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm**

**version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt**

**reading output11.csv ... [891 item(s), 977 transaction(s)] done [0.00s].**

**filtering, sorting and recoding items ... [9 item(s)] done [0.00s].**

**sorting and reducing transactions ... [50/977 transaction(s)] done [0.00s].**

**finding frequent item set(s) ... done [0.00s].**

**writing output\_rule\_11.out ... [0 rule(s)] done [0.00s].**

**My fpgrowth 是指我的程式碼跑出來的結果**

**Fpgrowth.exe 是指Christian Borgelt提供的執行檔測出來的數據**

* **兩者測出來的答案皆相同, 但在時間執行上面Christian Borgelt所提供的執行檔執行速度較快**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Compare different dataset** | | | | |
| **name** | **DATA 1** | **DATA 2** | **DATA 3** | **DATA 4** |
| **dataset** | **ntrans\_0.1**  **nitems\_0.1** | **ntrans\_0.1**  **nitems\_1** | **ntrans\_1**  **nitems\_0.1** | **ntrans\_1**  **nitems\_1** |
| **min\_support** | 5%(5) | 5%(5) | 5%(49) | 5%(49) |
| **min\_confidence** | 50% | 50% | 50% | 50% |
| **item** | 93 | 499 | 99 | 891 |
| **transaction** | 99 | 99 | 984 | 977 |
| **Item total count** | 967 | 1038 | 9701 | 10014 |
| **Support 數目 \* item** | 465 < Item total count | 2495 > Item total count | 4851 < Item total count | 43659 > Item total count |
| **Item total count / (Support 數目 \* item)** | 2.080 | 0.416 | 2.000 | 0.229 |
| **frequent item sets** | **287** | **33** | **187** | **9** |
| **rule** | **58** | **0** | **0** | **0** |

1. **support 個數 = transaction \* 5% ,因為我所設的 %數相同, 因此影響 support 個數 的是 transaction**
2. **如果support 個數 \* item < item total count ( item 的總共數量) , 表示這個 DATA 的 item 有很多能滿足大於 min\_support的數 , 而反之則表示這個 DATA 的 item 只有很少能滿足大於 min\_support的數 , 因此越多能滿足大於 min\_support的數的 frequent item sets 會有較多**

**從上述表格和 A 、 B 兩點看來, DATA 1、2、3、4 中, 他們的 “support 個數” 分別為 5、5、49、49, 而因為如果有越多能滿足support 的數量, 則frequent item sets 會有較多, 因此我認為影響 frequent item sets 的原因是support 個數 \* item 是否大於小於 item total count ( item 的總共數量)**

**而關於rule上的數量, 我認為首先必須滿足一點, 就是在 frequent item sets 中, 有較多包含兩個或以上 item 的 set, 或是含有單個 item 的 set的support值要大於min\_confidence值, 再來就是要符合 min\_confidence, 因此可見在 data2、data4中的frequent item sets, 沒有包含兩個或以上 item 的 set, 或是含有單個 item 的 set的support值大於min\_confidence值, 因此 rule 為0, 而在 data3 中就是confidence值沒有大於min\_confidence (data3 中 confidence最大值為 47% [63 <- 35(6.40244,47.7273)]), 因此 rule 為0**

|  |  |
| --- | --- |
| Kaggle(bakery)  (min\_support ≥ 5%, min\_confidence ≥ 50%) | |
| My fpgrowth | Fpgrowth.exe |
| {'Coffee'} 46.75754 | Coffee (46.7575) |
| {'Bread'} 31.98059 | Bread Coffee (8.79802) |
| {'Coffee', 'Bread'} 8.79802 | Bread (31.9806) |
| {'Tea'} 13.94052 | Tea (13.9405) |
| {'Cake'} 10.15076 | Cake Coffee (5.34903) |
| {'Cake', 'Coffee'} 5.34903 | Cake (10.1508) |
| {'Pastry'} 8.41594 | Pastry (8.41594) |
| {'Sandwich'} 7.02189 | Sandwich (7.02189) |
| {'Medialuna'} 6.04089 | Medialuna (6.04089) |
| {'Hot chocolate'} 5.70012 | Hot chocolate (5.70012) |
| {'Cookies'} 5.31805 | Cookies (5.31805) |

**fpgrowth -s5 -ts -c50 -f, outputbakery.csv output\_bakery.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm**

**version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt**

**reading outputbakery.csv ... [94 item(s), 9684 transaction(s)] done [0.00s].**

**filtering, sorting and recoding items ... [9 item(s)] done [0.00s].**

**sorting and reducing transactions ... [186/9684 transaction(s)] done [0.00s].**

**writing output\_bakery.out ... [11 set(s)] done [0.00s].**

**Association rules:**

**Coffee <- Cake(5.34903,52.6958)**

**這則 Association rules 表示在有買蛋糕的顧客中, 買咖啡的比例大約佔53%**

**而只有少少的一則規則的原因是因為 frequent item sets 少, 還有frequent item sets 的內容中僅有 2 個有包含兩個以上的 item (**{'Coffee', 'Bread'}、{'Cake', 'Coffee'} **), 而且在 frequent item sets 中本身的 support %數, 也都沒有 > 50%, 因此能滿足產生規則的不多, 像是** Coffee <- Bread **的confidence 只有約27%, 就不滿足min\_confidence > 50%**

**fpgrowth -s5 -tr -o -c50 -f, outputbakery.csv output\_bakery.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm**

**version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt**

**reading outputbakery.csv ... [94 item(s), 9684 transaction(s)] done [0.00s].**

**filtering, sorting and recoding items ... [9 item(s)] done [0.00s].**

**sorting and reducing transactions ... [186/9684 transaction(s)] done [0.00s].**

**finding frequent item set(s) ... done [0.00s].**

**writing output\_bakery.out ... [1 rule(s)] done [0.00s].**

|  |  |  |  |
| --- | --- | --- | --- |
| Kaggle(titanic)  (min\_support ≥ 12%, min\_confidence ≥ 70%) | | | |
| My fpgrowth | Fpgrowth.exe | My fpgrowth | Fpgrowth.exe |
| {'Embarked\_S'} 72.27834 | Embarked\_S (72.2783) | {'Pclass\_3', 'Survived\_0', 'Embarked\_S', 'Age(21-40)'} 14.25365 | Age(21-40) Pclass\_3 Survived\_0 (16.7228) |
| {'male'} 64.75870 | male Embarked\_S (49.4949) | {'male', 'Age(21-40)', 'Survived\_0', 'Pclass\_3'} 14.02918 | Age(21-40) Pclass\_3 (21.5488) |
| {'male', 'Embarked\_S'} 49.49495 | male (64.7587) | {'male', 'Survived\_0', 'Embarked\_S', 'Age(21-40)', 'Pclass\_3'} 12.00898 | Age(21-40) (43.2099) |
| {'Survived\_0'} 61.61616 | Survived\_0 male (52.5253) | {'Survived\_1'} 38.38384 | Survived\_1 Embarked\_S (24.3547) |
| {'Survived\_0', 'Embarked\_S'} 47.92368 | Survived\_0 Embarked\_S male (40.853) | {'Survived\_1', 'Embarked\_S'} 24.35466 | Survived\_1 Age(21-40) Embarked\_S (12.3457) |
| {'male', 'Survived\_0'} 52.52525 | Survived\_0 Embarked\_S (47.9237) | {'Survived\_1', 'male'} 12.23345 | Survived\_1 Age(21-40) (17.1717) |
| {'male', 'Survived\_0', 'Embarked\_S'} 40.85297 | Survived\_0 (61.6162) | {'Survived\_1', 'Age(21-40)'} 17.17172 | Survived\_1 Pclass\_3 (13.3558) |
| {'Pclass\_3'} 55.10662 | Pclass\_3 Survived\_0 (41.7508) | {'Survived\_1', 'Age(21-40)', 'Embarked\_S'} 12.34568 | Survived\_1 male (12.2334) |
| {'Pclass\_3', 'Embarked\_S'} 39.61841 | Pclass\_3 Embarked\_S Survived\_0 (32.0988) | {'Survived\_1', 'Pclass\_3'} 13.35578 | Survived\_1 (38.3838) |
| {'male', 'Pclass\_3'} 38.94501 | Pclass\_3 Embarked\_S (39.6184) | {'female'} 35.24130 | female Survived\_1 (26.1504) |
| {'male', 'Embarked\_S', 'Pclass\_3'} 29.74186 | Pclass\_3 male Survived\_0 (33.67) | {'female', 'Embarked\_S'} 22.78339 | female Embarked\_S Survived\_1 (15.7127) |
| {'Pclass\_3', 'Survived\_0'} 41.75084 | Pclass\_3 male Embarked\_S Survived\_0 (25.9259) | {'Pclass\_3', 'female'} 16.16162 | female Embarked\_S (22.7834) |
| {'Pclass\_3', 'Survived\_0', 'Embarked\_S'} 32.09877 | Pclass\_3 male Embarked\_S (29.7419) | {'female', 'Age(21-40)'} 15.26375 | female Pclass\_3 (16.1616) |
| {'male', 'Survived\_0', 'Pclass\_3'} 33.67003 | Pclass\_3 male (38.945) | {'Survived\_1', 'female'} 26.15039 | female Age(21-40) Survived\_1 (12.009) |
| {'male', 'Survived\_0', 'Embarked\_S', 'Pclass\_3'} 25.92593 | Pclass\_3 (55.1066) | {'Survived\_1', 'female', 'Embarked\_S'} 15.71268 | female Age(21-40) (15.2637) |
| {'Age(21-40)'} 43.20988 | Age(21-40) Embarked\_S (34.119) | {'Survived\_1', 'female', 'Age(21-40)'} 12.00898 | female (35.2413) |
| {'Age(21-40)', 'Embarked\_S'} 34.11897 | Age(21-40) male Embarked\_S (22.5589) | {'Pclass\_1'} 24.24242 | Pclass\_1 Survived\_1 (15.2637) |
| {'male', 'Age(21-40)'} 27.94613 | Age(21-40) male (27.9461) | {'Pclass\_1', 'Embarked\_S'} 14.25365 | Pclass\_1 Embarked\_S (14.2536) |
| {'male', 'Age(21-40)', 'Embarked\_S'} 22.55892 | Age(21-40) Survived\_0 Embarked\_S (21.7733) | {'male', 'Pclass\_1'} 13.69248 | Pclass\_1 male (13.6925) |
| {'Age(21-40)', 'Survived\_0'} 26.03816 | Age(21-40) Survived\_0 male Embarked\_S (18.9675) | {'Survived\_1', 'Pclass\_1'} 15.26375 | Pclass\_1 (24.2424) |
| {'Survived\_0', 'Embarked\_S', 'Age(21-40)'} 21.77329 | Age(21-40) Survived\_0 male (22.7834) | {'Pclass\_2'} 20.65095 | Pclass\_2 Embarked\_S (18.4063) |
| {'male', 'Age(21-40)', 'Survived\_0'} 22.78339 | Age(21-40) Survived\_0 (26.0382) | {'Embarked\_S', 'Pclass\_2'} 18.40629 | Pclass\_2 male (12.1212) |
| {'male', 'Survived\_0', 'Embarked\_S', 'Age(21-40)'} 18.96745 | Age(21-40) Pclass\_3 Embarked\_S (18.2941) | {'male', 'Pclass\_2'} 12.12121 | Pclass\_2 (20.651) |
| {'Pclass\_3', 'Age(21-40)'} 21.54882 | Age(21-40) Pclass\_3 male Embarked\_S (13.9169) | {'Embarked\_C'} 18.85522 | Embarked\_C (18.8552) |
| {'Pclass\_3', 'Age(21-40)', 'Embarked\_S'} 18.29405 | Age(21-40) Pclass\_3 male (16.3861) | {'Age(41-65)'} 15.93715 | Age(41-65) Embarked\_S (12.1212) |
| {'male', 'Age(21-40)', 'Pclass\_3'} 16.38608 | Age(21-40) Pclass\_3 Survived\_0 Embarked\_S (14.2536) | {'Embarked\_S', 'Age(41-65)'} 12.12121 | Age(41-65) (15.9371) |
| {'male', 'Age(21-40)', 'Embarked\_S', 'Pclass\_3'} 13.91695 | Age(21-40) Pclass\_3 Survived\_0 male Embarked\_S (12.009) | {'Age(10-20)'} 12.90685 | Age(10-20) (12.9068) |
| {'Pclass\_3', 'Age(21-40)', 'Survived\_0'} 16.72278 | Age(21-40) Pclass\_3 Survived\_0 male (14.0292) |  |  |

**fpgrowth -s12 -ts -o -c70 -f, outputtitanic.csv output\_titanic.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm**

**version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt**

**reading outputtitanic.csv ... [16 item(s), 891 transaction(s)] done [0.00s].**

**filtering, sorting and recoding items ... [12 item(s)] done [0.00s].**

**sorting and reducing transactions ... [94/891 transaction(s)] done [0.00s].**

**writing output\_titanic.out ... [55 set(s)] done [0.00s].**

|  |  |
| --- | --- |
| Association rules  (min\_support ≥ 12%, min\_confidence ≥ 70%) | |
| Embarked\_S <- (72.2783,72.2783) | Embarked\_S <- Age(21-40) Survived\_0(21.7733,83.6207) |
| Embarked\_S <- male(49.4949,76.4298) | Embarked\_S <- Age(21-40) Survived\_0 male(18.9675,83.2512) |
| Embarked\_S <- Survived\_0(47.9237,77.7778) | male <- Age(21-40) Survived\_0 Embarked\_S(18.9675,87.1134) |
| Embarked\_S <- Survived\_0 male(40.853,77.7778) | Survived\_0 <- Age(21-40) male Embarked\_S(18.9675,84.0796) |
| male <- Survived\_0 Embarked\_S(40.853,85.2459) | male <- Age(21-40) Survived\_0(22.7834,87.5) |
| Survived\_0 <- male Embarked\_S(40.853,82.5397) | Survived\_0 <- Age(21-40) male(22.7834,81.5261) |
| male <- Survived\_0(52.5253,85.2459) | Embarked\_S <- Age(21-40) Pclass\_3(18.2941,84.8958) |
| **Survived\_0 <- male(52.5253,81.1092)** | Embarked\_S <- Age(21-40) Pclass\_3 male(13.9169,84.9315) |
| Embarked\_S <- Pclass\_3(39.6184,71.8941) | male <- Age(21-40) Pclass\_3 Embarked\_S(13.9169,76.0736) |
| Embarked\_S <- Pclass\_3 male(29.7419,76.3689) | male <- Age(21-40) Pclass\_3(16.3861,76.0417) |
| male <- Pclass\_3 Embarked\_S(29.7419,75.0708) | Embarked\_S <- Age(21-40) Pclass\_3 Survived\_0(14.2536,85.2349) |
| male <- Pclass\_3(38.945,70.6721) | Survived\_0 <- Age(21-40) Pclass\_3 Embarked\_S(14.2536,77.9141) |
| Embarked\_S <- Pclass\_3 Survived\_0(32.0988,76.8817) | Embarked\_S <- Age(21-40) Pclass\_3 Survived\_0 male(12.009,85.6) |
| Survived\_0 <- Pclass\_3 Embarked\_S(32.0988,81.0198) | male <- Age(21-40) Pclass\_3 Survived\_0 Embarked\_S(12.009,84.252) |
| Embarked\_S <- Pclass\_3 Survived\_0 male(25.9259,77) | Survived\_0 <- Age(21-40) Pclass\_3 male Embarked\_S(12.009,86.2903) |
| male <- Pclass\_3 Survived\_0 Embarked\_S(25.9259,80.7692) | male <- Age(21-40) Pclass\_3 Survived\_0(14.0292,83.8926) |
| Survived\_0 <- Pclass\_3 male Embarked\_S(25.9259,87.1698) | Survived\_0 <- Age(21-40) Pclass\_3 male(14.0292,85.6164) |
| male <- Pclass\_3 Survived\_0(33.67,80.6452) | Survived\_0 <- Age(21-40) Pclass\_3(16.7228,77.6042) |
| Survived\_0 <- Pclass\_3 male(33.67,86.4553) | Embarked\_S <- Survived\_1 Age(21-40)(12.3457,71.8954) |
| **Survived\_0 <- Pclass\_3(41.7508,75.7637)** | Survived\_1 <- female Age(21-40)(12.009,78.6765) |
| Embarked\_S <- Age(21-40)(34.119,78.961) | **Survived\_1 <- female(26.1504,74.2038)** |
| Embarked\_S <- Age(21-40) male(22.5589,80.7229) | Embarked\_S <- Pclass\_2(18.4063,89.1304) |
|  | Embarked\_S <- Age(41-65)(12.1212,76.0563) |

**Pclass：乘客的艙位等級(1、2、3 )**

**Embarked : 登船港口( C、Q、S )**

**Sex：乘客性別(male、female)**

**Age：乘客年齡分布( 0-10、11-20、21-40、40-65、65- )**

**Survived：沉船之後，乘客是否存活( 0 : 死亡、1 : 存活 )**

**如果要分析男性、女性乘客, 在沉船之後, 是否存活的話, 就要找出箭頭指向Survived\_0或Survived\_1 的規則, 並且另一端有男性女性的判斷資訊(Sex)**

* **Survived\_0 <- male(52.5253,81.1092)**

**Survived\_1 <- female(26.1504,74.2038)**

**因此可以發現在乘客為男性的情況下, 在沉船之後, 有很高的比例會死亡; 而在乘客為女性的情況下, 在沉船之後, 有很高的比例會存活**

**如果要分析乘客乘坐的艙位等級, 在沉船之後, 是否存活的話, 就要找出箭頭指向Survived\_0或Survived\_1 的規則, 並且另一端有艙位等級的判斷資訊(Pclass)**

* **Survived\_0 <- Pclass\_3(41.7508,75.7637)**

**因此可以發現在**Pclass 為3**的情況下, 在沉船之後, 有很高的比例會死亡**

**fpgrowth -s12 -tr -v(%S,%C) -o -c70 -f, outputtitanic.csv output\_rule\_titanic.out**

**fpgrowth - find frequent item sets with the fpgrowth algorithm**

**version 6.17 (2017.05.30) (c) 2004-2017 Christian Borgelt**

**reading outputtitanic.csv ... [16 item(s), 891 transaction(s)] done [0.00s].**

**filtering, sorting and recoding items ... [12 item(s)] done [0.00s].**

**sorting and reducing transactions ... [94/891 transaction(s)] done [0.00s].**

**finding frequent item set(s) ... done [0.00s].**

**writing output\_rule\_titanic.out ... [45 rule(s)] done [0.00s].**