SEQUENCE ANALYSIS

The dataset of interest here (FaultCodes.Rdata) involves fault codes from a fleet of over 15,000 vehicles. Each vehicle is identified by its own unique *vehicleID*. The fault codes registered by the vehicle are given in the variable *code*, and the order in which these codes were registered is provided in the *sequence* variable. For each vehicleID, the total number of fault codes registered is also provided as *numFaults*. The first step is to load this data into R:

```
> load('/Users/shaina/Library/Mobile Documents/com~apple~CloudDocs/Data Mining 2017/Data Sets for Demo/Fault Codes
```

In order to start mining sequence rules right away, use the seqecreate from the TraMineR package to create an event sequence object that serves as a starting place for mining. From there, we have to subset the rules that we're interested in, otherwise we're just taking a long form to represent our data. Finally, to get the same output that we receive from enterprise miner, we have to use the undocumented function TraMineR:::seqerules. The three colons in this function indicate that it is not currently a public, documented function. Use it at your own risk!

```
> library(TraMineR)
> FCseqdata = seqecreate(id=FC$vehicleID, timestamp=FC$sequence, event=FC$Code)
> FCsubseqdata <- seqefsub(FCseqdata, pMinSupport=0.005)
> rules = TraMineR:::seqerules(FCsubseqdata)
> head(rules)
```

```
              Rules Support       Conf      Lift Standardlift  JMeasure
1 (C7610) => (C9245)    2501 0.52641549 3.1666854   0.99800475 0.4889615
2 (A1002) => (C5151)    2429 0.54843080 3.3953159   0.99753589 0.5640131
3 (B2953) => (F5722)    2390 0.50464527 3.1777475   0.99832912 0.4633112
4 (C7610) => (C7610)     455 0.09576931 0.3038776   0.09575041 0.1979494
5 (B2953) => (B2953)     440 0.09290541 0.2957240   0.09288650 0.2026118
6 (B2953) => (C7610)     439 0.09269426 0.2941204   0.09267529 0.2045461
  ImplicStat p.value p.valueB1 p.valueB2
1  -27.20471       0         0         0
2  -28.11981       0         0         0
3  -25.96553       0         0         0
4  -57.04109       0         0         0
5  -56.99234       0         0         0
6   11.52897       1         1         1
```

```
>
```

The values of support output by this function are simply the number of vehicles that exhibit that sequence rule. To match the output of SAS EM, we'd probably want to convert this to a probability/percentage. We can then order the rules by whatever statistic we're most interested in, usually lift.

```
> rules$Support = rules$Support/15075
> #rules=rules[order(-rules$Support),]
> #rules=rules[order(-rules$Conf),]
> rules=rules[order(-rules$Lift),]
> rules[1:10,]
```

```
                         Rules     Support       Conf     Lift Standardlift
35          (H0009) => (H8780) 0.015588723 0.11709018 5.825526    0.7755756
7           (A1034) => (D3937) 0.028192371 0.15238437 5.317579    0.9837961
28          (F1111) => (C8073) 0.017247098 0.11139674 5.058150    0.7831303
34          (H0009) => (D1103) 0.015588723 0.11709018 4.903151    0.6527747
15          (A6409) => (F7142) 0.024610282 0.14845938 4.875872    0.8082768
22          (C9876) => (A1163) 0.018839138 0.15130527 4.344623    0.5409486
24          (F1111) => (D1991) 0.018242123 0.11782348 4.300700    0.6658561
73  (A1002) => (C5151)-(A1002) 0.012404643 0.04222172 3.403703    1.0000000
189 (A1002) => (C5151)-(C9876) 0.006102819 0.02077218 3.403703    1.0000000
201 (A1002) => (C5151)-(F1111) 0.005970149 0.02032061 3.403703    1.0000000
      JMeasure ImplicStat      p.value    p.valueB1    p.valueB2
35  0.16492631 -4.4571246 4.153317e-06 9.133143e-04 7.182673e-04
7   0.20074692 -6.6298645 1.679978e-11 3.695952e-09 2.906363e-09
28  0.13765454 -4.4079753 5.217072e-06 1.147100e-03 9.021485e-04
34  0.14073297 -4.3168909 7.912115e-06 1.739158e-03 1.367865e-03
15  0.17988244 -6.0928954 5.544325e-10 1.219751e-07 9.591682e-08
22  0.16318100 -5.4420595 2.633404e-08 5.793472e-06 4.555778e-06
24  0.12377102 -4.6606440 1.576108e-06 3.466839e-04 2.726297e-04
73  0.03224928 -1.9967716 2.292500e-02 9.939167e-01 9.819063e-01
189 0.01570017 -0.9792496 1.637283e-01 1.000000e+00 1.000000e+00
201 0.01535548 -0.9578976 1.690572e-01 1.000000e+00 1.000000e+00
```