348 Final Project

Shay Lebovitz

11/19/2020

For my project, I will be analyzing unemployment duration data from the R package `Ecdat`. It contains 3343 observations of 11 variables. These 11 variables are: 1. `spell`: length of unemployment spell in number of two-week intervals. 2. `censor1`: dummy variable equal to 1 if re-employed at a full-time job. 3. `censor2`: dummy variable equal to 1 if re-employed at a part-time job. 4. `censor3`: dummy variable equal to 1 if re-employed but left job, part/full-time status unknown. 5. `censor4`: dummy variable equal to 1 if still jobless. 6. `age`: age of person. 7. `ui`: dummy variable equal to one if person filed unemployment insurance claim. 8. `reprate`: eligible replacement rate. 9. `disrate`: eligible disregard rate. 10. `logwage`: log weekly earnings in lost job (1985$). 11. `tenure`: years tenure in lost job.

The goal is to determine if linear discriminant analysis can adequately separate the four censor groups based on information in the `spell`, `age`, `reprate`, `disrate`, `logwage`, and `tenure` variables. This will be more complicated than the examples given in class, which only dealt with 2 distinct groups. If the analysis doesn't work, I may combine the three re-emloyment variables into one, simply called `re-employed`, and compare that to `censor4`, which I'd call `unemployed`. Since I am not sure how dummy variables will work in LDA, I will leave `ui` out of the analysis.

```
head(UnempDur)
```

```
##   spell censor1 censor2 censor3 censor4 age  ui reprate disrate logwage tenure
## 1     5       1       0       0       0  41  no   0.179   0.045 6.89568      3
## 2    13       1       0       0       0  30 yes   0.520   0.130 5.28827      6
## 3    21       1       0       0       0  36 yes   0.204   0.051 6.76734      1
## 4     3       1       0       0       0  26 yes   0.448   0.112 5.97889      3
## 5     9       0       0       1       0  22 yes   0.320   0.080 6.31536      0
## 6    11       0       0       0       1  43 yes   0.187   0.047 6.85435      9
```

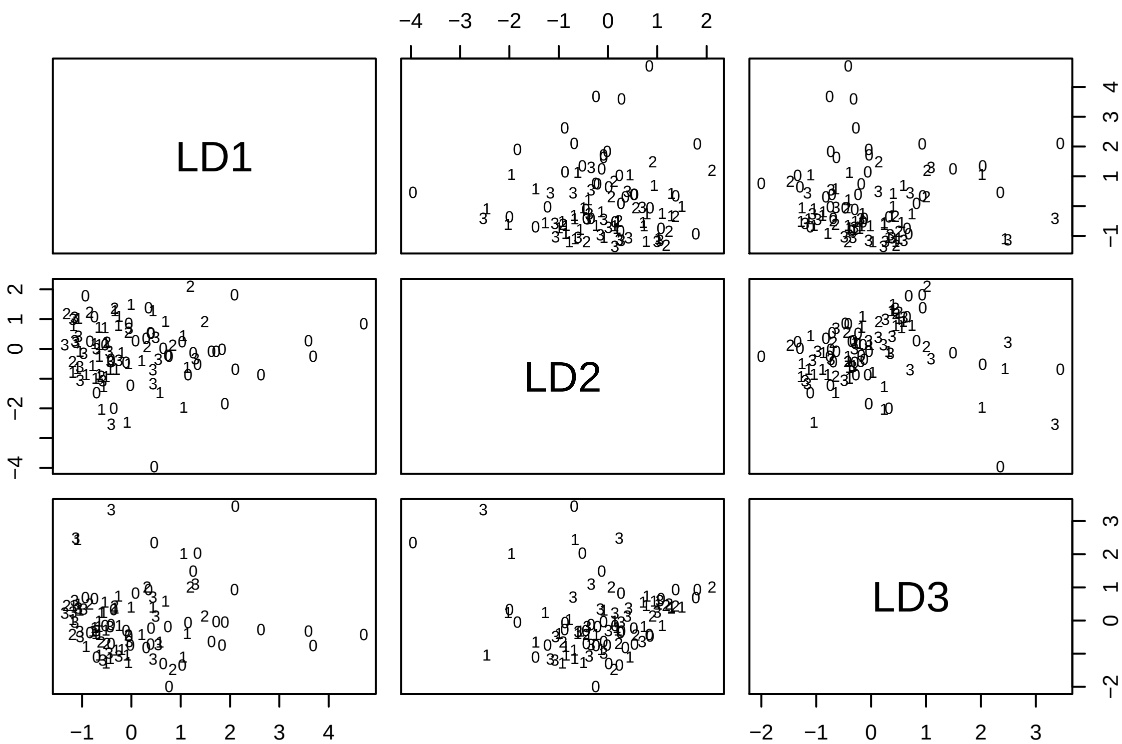
For the sake of graphics and run-time, I will reduce the number of observations to 100, sampled randomly from the data set. Furthermore, it should be noted that a 1 refers to `censor1` (re-employed full-time), a 2 refers to `censor2` (re-employed part-time), a 3 refers to `censor3` (re-employed but left job), and a 0 refers to `censor4` (still jobless).

```
unemp_lda
```

```
## Call:
## lda(status ~ spell + age + reprate + disrate + logwage + tenure,
##     data = unemp)
##
## Prior probabilities of groups:
##    0    1    2    3
## 0.34 0.34 0.11 0.21
##
## Group means:
##      spell      age   reprate   disrate  logwage   tenure
## 0 8.617647 41.35294 0.4627059 0.1042353 5.657338 5.323529
## 1 3.558824 36.08824 0.4771765 0.1123529 5.649579 4.117647
```

```
## 2 5.000000 32.45455 0.4636364 0.1058182 5.445303 1.727273
## 3 3.142857 35.19048 0.4768095 0.1125714 5.631005 5.142857
##
## Coefficients of linear discriminants:
##                 LD1         LD2         LD3
## spell    0.19698549  0.02478073  0.02431460
## age      0.04291352 -0.02396157 -0.05142623
## reprate -0.01227907 -8.58152762 -3.36692881
## disrate -0.37866216 -2.14037928 -1.57976809
## logwage -0.14892938 -1.88998858 -0.98240017
## tenure  -0.02711305 -0.06381388  0.16088699
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.8346 0.1495 0.0159
```

```r
plot(unemp_lda)
```



Due to the issue of the number of groups being larger than 2, there are multiple linear discriminant functions. But we are clearly able to see that there is no clear distinction between the groups.

To see numerically how many of each class the classification prediction got right, we can refer to this table:

```r
table(unemp$status, unemp_pred$class, dnn = c("actual", "predicted"))
```

```
##       predicted
## actual  0  1  2  3
##      0 21 12  1  0
##      1  6 27  0  1
##      2  4  7  0  0
##      3  5 14  0  2
```

We see that the functions do a decent job of discriminating group 0 (jobless), with a success rate of 21/34 =

62%. Most that were predicted wrong were predicted as group 1, which is strange because group 1 (full-time re-employment) is the least similar to group 0. The function does a better job identifying group 1, with a $27/34 = 79\%$ success rate. Most group 1's were incorrectly identified as group 0, which is strange as explained above. However, the function does a very poor job classifying groups 2 (part-time re-employment) and group 3 (re-employed but left job), with a $0/11 = 0\%$ and $2/21 = 10\%$ success rate. Both of these groups were mostly wrongly predicted as belonging to group 1. Based on this analysis, I don't think linear discriminant analysis is a useful technique for this data set. The confusion matrix shown above has drawbacks, however. It uses the data for subject j to estimate the coefficients for the confusion matrix. By using cross-validation, we can eliminate this problem and thus provide a better estimate of "out of sample" performance.

```
table(unemp$status, unemp_lda_cv$class, dnn = c('actual', 'predicted'))
```

```
##       predicted
## actual  0  1  2  3
##      0 17 14  2  1
##      1  6 25  2  1
##      2  4  7  0  0
##      3  5 16  0  0
```

We see that the accuracy only went down using the cross-validaiton method. The success rate of group 0 dropped to $17/34 = 50\%$ and group 1 dropped to $25/34 = 73.5\%$. Group 2 stayed at a 0% success rate and group 3 dropped to 0% as well.

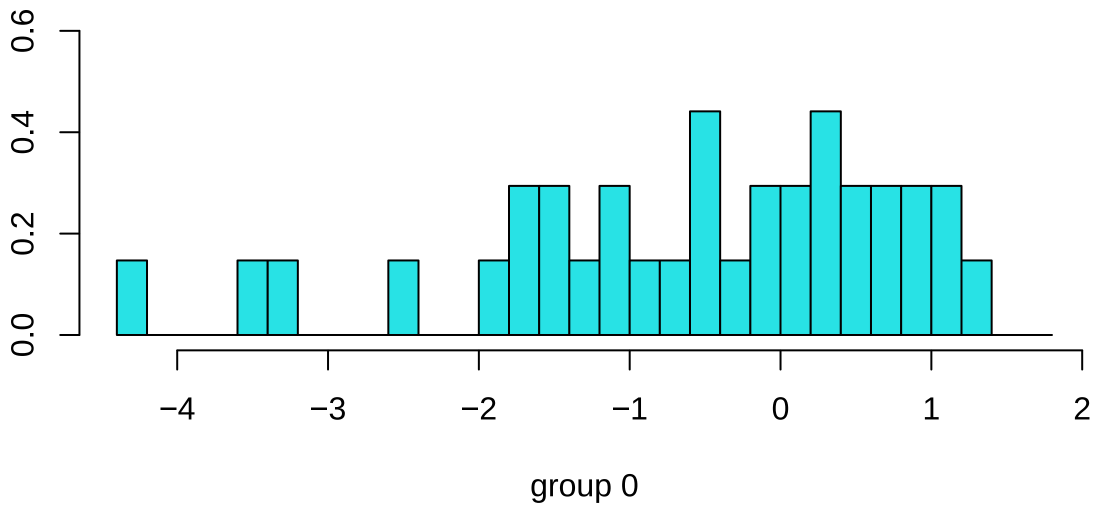
From now on, all analyses will be performed with the cross-validation method.

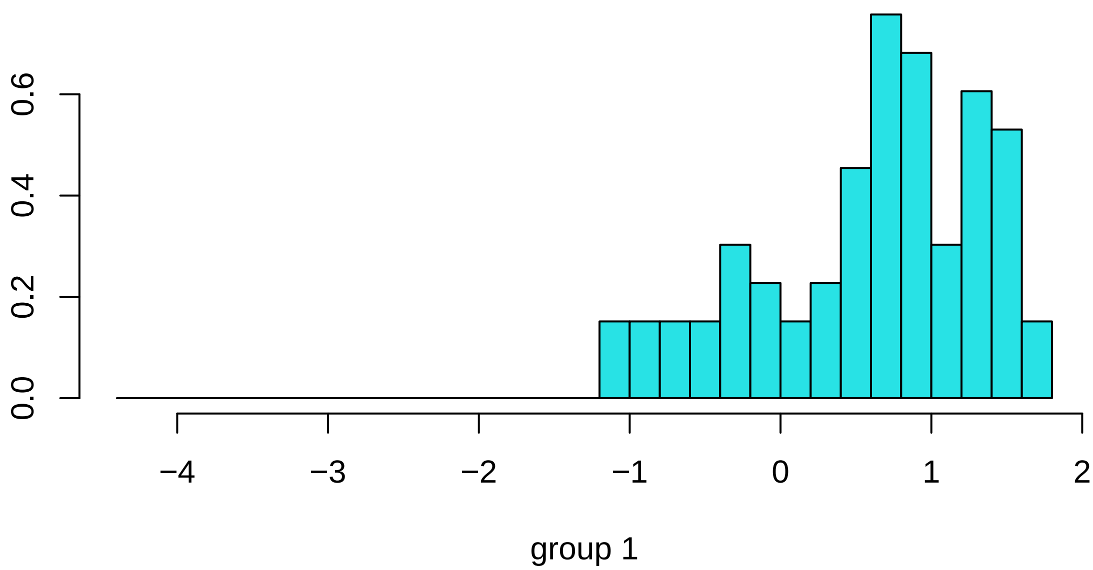
Now, I want to now try a simpler problem and see if LDA is still of any use. I will combine groups 1, 2, and 3 into a `re-employed` variable (now group 1) and run the analysis against group 0, the `unemployed` variable.

```
unemp2_lda
```

```
## Call:
## lda(status2 ~ spell + age + reprate + disrate + logwage + tenure,
##     data = unemp2)
##
## Prior probabilities of groups:
##    0    1
## 0.34 0.66
##
## Group means:
##      spell      age   reprate   disrate  logwage   tenure
## 0 8.617647 41.35294 0.4627059 0.1042353 5.657338 5.323529
## 1 3.666667 35.19697 0.4748030 0.1113333 5.609623 4.045455
##
## Coefficients of linear discriminants:
##                  LD1
## spell   -0.195994175
## age     -0.044191313
## reprate -0.638542322
## disrate  0.229578725
## logwage  0.009982737
## tenure   0.019347746
```

```r
plot(unemp2_lda)
```





While group 1 is almost entirely concentrated above -1, group 0 is much more spread out. However, the clear distinction between the two groups simply isn't there, so again LDA is not a very appropriate method. I will still analyze the table results, using the cross-validation method described above:

```r
table(unemp2$status2, unemp2_lda_cv$class, dnn = c('actual', 'predicted'))
```

```
##          predicted
## actual    0  1
##         0 13 21
##         1  7 59
```

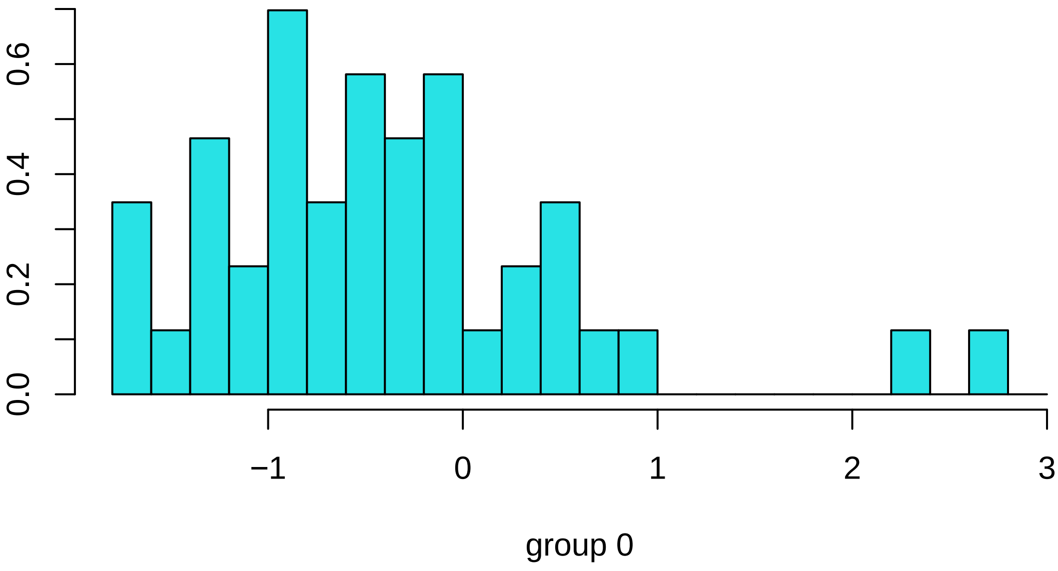
The top confusion matrix The method does a poor job of classifying group 0 (jobless) with a 13/34 = 38.2% success rate. However, it does a fairly good job of classifying group 1 (re-employed) with a 59/66 =

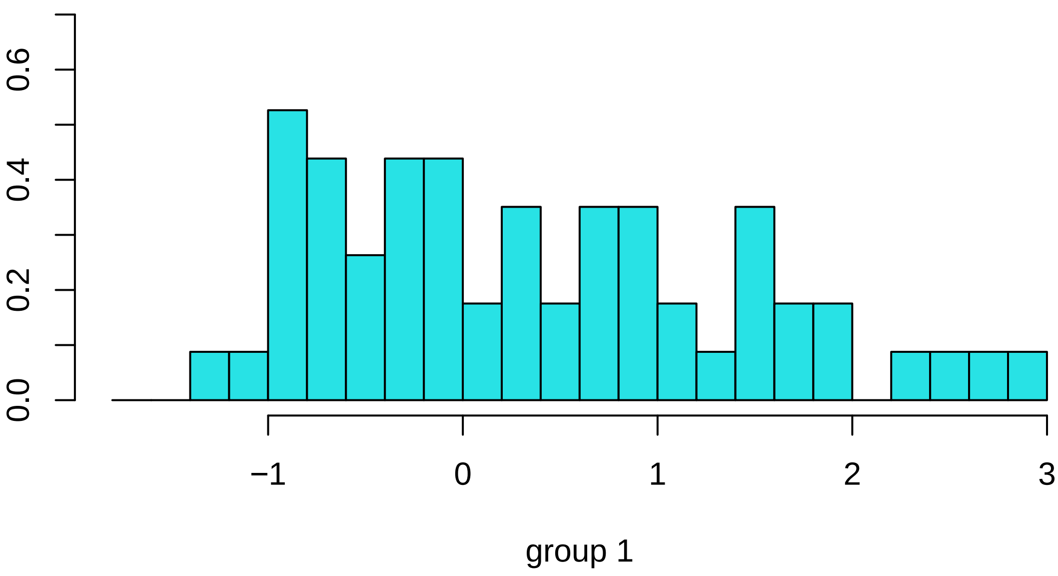
89.4% success rate. Overall, it still looks as though LDA is not appropriate in this case in determining the re-employment status of temporarily unemployed people using unemployment duration, replacement rate, disregard rate, wage, or tenure at previous job. I will attempt one more LDA, this time with the groups being whether or not the person filed for unemployment insurance.

```
unemp_lda_ui
```

```
## Call:
## lda(ui ~ spell + age + reprate + disrate + logwage + tenure,
##     data = unemp)
##
## Prior probabilities of groups:
##    0    1
## 0.43 0.57
##
## Group means:
##      spell      age   reprate   disrate  logwage   tenure
## 0 3.790698 33.90698 0.4789302 0.1182093 5.566305 3.139535
## 1 6.526316 39.84211 0.4644737 0.1019123 5.670763 5.491228
##
## Coefficients of linear discriminants:
##                 LD1
## spell    0.13807182
## age      0.04479743
## reprate  2.13086437
## disrate -3.56138941
## logwage  0.29691040
## tenure   0.02872657
```

```
plot(unemp_lda_ui)
```





We see that there is clear differentiation at the fringes, with group 0 (no unemployment insurance claim) containing most of the values less than -1, and group 1 (unemployment insurance claim) containing most of the values greather than 1. However, there is clearly lots of overlap in between -1 and 1. Again, this shows that the method is not very accurate. We can look at the table (using cross-validation) to numerically determine the accuracy of the analysis:

```
table(unemp$ui, unemp_lda_ui_cv$class, dnn = c('actual', 'predicted'))
```

```
##       predicted
## actual  0  1
##      0 22 21
##      1 19 38
```

Here, we see that it correctly predicted group 0 22/43 = 51.2% of the time, and predicted group 1 38/57 = 66.7% of the time. Again, these numbers are simply not high enough for LDA to be deemed an appropriate method of classifying re-employment or unemployment insurance filing status based on this data set.

Quadratic discriminant analysis (QDA) could be used instead of linear discriminant analysis. This method allows for more flexibility and could perhaps lead to better classification results. Unlike LDA, QDA does not require the assumption that the within-group covariance matrices of $\mathbf{X}$ are equal.

```
cov(unemp2_0)
```

```
##               spell          age      reprate      disrate      logwage
## spell   42.78877005   6.10873440 -0.057206774  0.046516934  0.123125134
## age      6.10873440 142.17468806 -0.323680927  0.043368984  2.175352674
## reprate -0.05720677  -0.32368093  0.010356578  0.002255799 -0.038737114
## disrate  0.04651693   0.04336898  0.002255799  0.005358367 -0.007644404
## logwage  0.12312513   2.17535267 -0.038737114 -0.007644404  0.279157000
## tenure   4.76381462  41.63992870 -0.142477718 -0.047866310  1.370970481
##              tenure
## spell    4.76381462
## age     41.63992870
## reprate -0.14247772
## disrate -0.04786631
## logwage  1.37097048
## tenure  57.55882353
```

```
cov(unemp2_1)
```

```
##                spell         age      reprate      disrate     logwage
## spell    9.517948718  -0.4256410 -0.003851282 -0.024856410 -0.14870641
## age     -0.425641026 119.1759907 -0.311683683 -0.263528205  1.84501903
## reprate -0.003851282  -0.3116837  0.009236776  0.002755205 -0.02784437
## disrate -0.024856410  -0.2635282  0.002755205  0.004373703 -0.01185040
## logwage -0.148706410   1.8450190 -0.027844375 -0.011850399  0.22266467
## tenure   0.584615385  32.5755245 -0.113160140 -0.030507692  0.52800015
##              tenure
## spell    0.58461538
## age     32.57552448
## reprate -0.11316014
## disrate -0.03050769
## logwage  0.52800015
## tenure  35.55174825
```

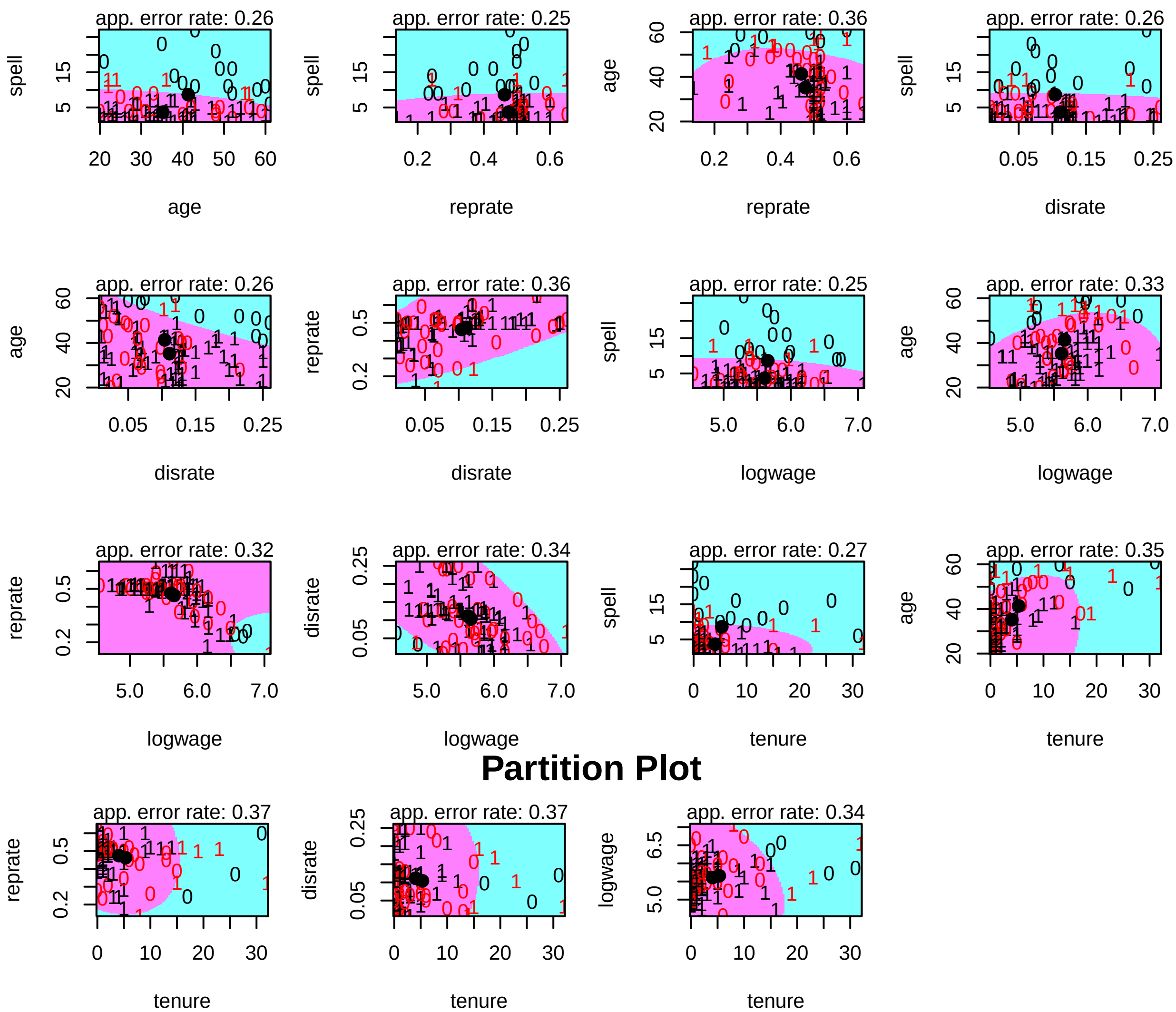
The matrices above show the covariance matrices of two groups: group 0 (still unemployed) and group 1 (re-employed). We see thay many of the values are relatively close, but some differ by an order of magnitude or more. Likewise, some values are positive in one group and negative in another. This suggests that the assumption of LDA that the covariance matrices of the two groups are equal may not hold very well. Thus, a quadratic discriminant analysis will be performed on this data set, to see if it produces any better results than LDA did.

```
unemp2_qda
```

```
## Call:
## qda(status2 ~ spell + age + reprate + disrate + logwage + tenure,
##     data = unemp2)
##
## Prior probabilities of groups:
```

```
##    0    1
## 0.34 0.66
##
## Group means:
##      spell      age   reprate   disrate  logwage   tenure
## 0 8.617647 41.35294 0.4627059 0.1042353 5.657338 5.323529
## 1 3.666667 35.19697 0.4748030 0.1113333 5.609623 4.045455
```

```r
partimat(as.factor(status2) ~ spell + age + reprate + disrate + logwage + tenure,
         data = unemp2, method = 'qda')
```




```r
table(unemp2$status2, unemp2_qda_cv$class, dnn = c('actual', 'predicted'))
```

```
##        predicted
## actual  0  1
##       0 14 20
##       1 11 55
```

We see that there is no clear distinctions between the 1's and 0's in any of the graphs, and thus even a quadratic function cannot distinguish them very well. For each combination of variables, we see error rates in the range of 0.25 - 0.40, most hovering around 0.35. The table shows us that the method accurately classified group 0 (still unemployed) 14/34 = 41.2% of the time, and group 1 (re-employed) 55/66 = 83.3% of the time.

This is a slightly better success rate for group 0 and a slightly worse success rate for group 1 than linear discriminant analysis. So, QDA does not seem any more appropriate than LDA for this classification problem. One might conclude from this analysis that tenure, unemployment spell, age, wage, replacement rate and disregard rate are not great predictors of the eventual re-employment status of workers.

Other multivariate analysis techniques could still be used to gain information from this data set, most notably Prinipal Components Analysis or Factor Analysis. These would allow the analyst to determine which combinations of variables or underlying factors are the most associated with variability in the re-employment status. However, no further analysis will be done on this project.

#R Output

```
#First LDA analysis - four groups
set.seed(12345)
unemp<- tibble(UnempDur)
unemp <- unemp %>%
  mutate(status = censor1 + 2*censor2 + 3*censor3 + 0*censor4)
unemp <- sample_n(unemp, 100)
unemp_lda <- lda(status ~ spell + age + reprate + disrate + logwage + tenure, data = unemp)
unemp_lda_cv <- lda(status ~ spell + age + reprate + disrate + logwage + tenure,
                    data = unemp, CV = T)
unemp_lda
```

```
## Call:
## lda(status ~ spell + age + reprate + disrate + logwage + tenure,
##     data = unemp)
##
## Prior probabilities of groups:
##    0    1    2    3
## 0.34 0.34 0.11 0.21
##
## Group means:
##      spell      age   reprate   disrate  logwage   tenure
## 0 8.617647 41.35294 0.4627059 0.1042353 5.657338 5.323529
## 1 3.558824 36.08824 0.4771765 0.1123529 5.649579 4.117647
## 2 5.000000 32.45455 0.4636364 0.1058182 5.445303 1.727273
## 3 3.142857 35.19048 0.4768095 0.1125714 5.631005 5.142857
##
## Coefficients of linear discriminants:
##                 LD1         LD2         LD3
## spell    0.19698549  0.02478073  0.02431460
## age      0.04291352 -0.02396157 -0.05142623
## reprate -0.01227907 -8.58152762 -3.36692881
## disrate -0.37866216 -2.14037928 -1.57976809
## logwage -0.14892938 -1.88998858 -0.98240017
## tenure  -0.02711305 -0.06381388  0.16088699
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.8346 0.1495 0.0159
```

```
unemp_lda_cv
```

```
## $class
##   [1] 1 1 1 0 2 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 3 1 1 0 1 0 1 1 3 1 0 0 0 0 1 1
##  [38] 1 0 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 2 0 0 1 0 1 0 0 1 0
##  [75] 1 1 0 1 2 1 0 2 1 0 1 1 0 0 1 1 1 0 0 0 1 0 1 1 0 1
## Levels: 0 1 2 3
##
## $posterior
##             0           1           2            3
## 1   0.14641998 0.397218516 0.193918737 0.2624427626
## 2   0.07178599 0.480602813 0.119178748 0.3284324488
## 3   0.10678120 0.472021217 0.135558266 0.2856393194
## 4   0.75012008 0.138112355 0.047510398 0.0642571616
## 5   0.05500348 0.324987829 0.368338432 0.2516702602
```

```
## 6   0.26515252 0.377516183 0.116047757 0.2412835415
## 7   0.26475967 0.395365018 0.131936643 0.2079386727
## 8   0.30232792 0.377779894 0.146079452 0.1738127330
## 9   0.19100850 0.298117211 0.284669897 0.2262043962
## 10  0.36107268 0.377806370 0.114553404 0.1465675417
## 11  0.18612502 0.341781903 0.246928746 0.2251643279
## 12  0.08989342 0.576173926 0.035676026 0.2982566236
## 13  0.15241772 0.433776402 0.127491639 0.2863142416
## 14  0.17592193 0.422574708 0.136348379 0.2651549789
## 15  0.18807597 0.485772155 0.110059069 0.2160928083
## 16  0.14624876 0.478213216 0.117547005 0.2579910204
## 17  0.15244830 0.486580584 0.024901309 0.3360698031
## 18  0.44532609 0.341119460 0.080806754 0.1327476929
## 19  0.08369811 0.400025289 0.190133171 0.3261434311
## 20  0.96169611 0.017148032 0.016140452 0.0050154100
## 21  0.17228076 0.564127771 0.053263562 0.2103279116
## 22  0.09632065 0.305817695 0.069514252 0.5283474008
## 23  0.16719658 0.453175258 0.025432370 0.3541957906
## 24  0.14534976 0.500704754 0.085851103 0.2680943807
## 25  0.34468112 0.340292741 0.156620235 0.1584059063
## 26  0.22157841 0.459334086 0.071614453 0.2474730520
## 27  0.68415485 0.136244162 0.016959766 0.1626412263
## 28  0.10142594 0.455459603 0.104590260 0.3385242007
## 29  0.15743870 0.452791111 0.100136159 0.2896340318
## 30  0.17788079 0.377117824 0.002706921 0.4422944671
## 31  0.21353005 0.379163565 0.209657442 0.1976489392
## 32  0.42516391 0.396706458 0.044775460 0.1333541754
## 33  0.98936967 0.002646678 0.007438313 0.0005453358
## 34  0.54426068 0.298746786 0.049338113 0.1076544199
## 35  0.37455506 0.314483476 0.093225006 0.2177364549
## 36  0.17933263 0.503448111 0.059527696 0.2576915634
## 37  0.08421217 0.410424894 0.230618144 0.2747447900
## 38  0.27631012 0.346634677 0.138555959 0.2384992430
## 39  0.48106160 0.184879713 0.195531341 0.1385273441
## 40  0.23948511 0.433925785 0.084576569 0.2420125349
## 41  0.16006176 0.512914931 0.040307918 0.2867153936
## 42  0.15070186 0.464647081 0.057711625 0.3269394329
## 43  0.12354165 0.577735469 0.124210001 0.1745128835
## 44  0.22363457 0.302172170 0.213932879 0.2602603812
## 45  0.19560387 0.481677287 0.072067284 0.2506515632
## 46  0.15296881 0.447102006 0.128451334 0.2714778483
## 47  0.09963652 0.476140722 0.161782575 0.2624401835
## 48  0.25085294 0.466309336 0.046988647 0.2358490719
## 49  0.72107245 0.150410578 0.061583107 0.0669338692
## 50  0.16620875 0.368050129 0.190702173 0.2750389455
## 51  0.08839085 0.381662997 0.243512565 0.2864335863
## 52  0.42835765 0.301102846 0.140269290 0.1302702157
## 53  0.11913008 0.422958190 0.149223306 0.3086884189
## 54  0.10519276 0.561575653 0.052181149 0.2810504347
## 55  0.36721874 0.228119863 0.253157273 0.1515041185
## 56  0.10133411 0.435647845 0.158371537 0.3046465101
## 57  0.27588618 0.359443105 0.178846585 0.1858241292
## 58  0.13577541 0.479357683 0.076951916 0.3079149856
## 59  0.08716332 0.487817997 0.061911447 0.3631072375
```

```
## 60  0.47778249 0.316848685 0.034652525 0.1707162964
## 61  0.11890939 0.504014092 0.061621450 0.3154550718
## 62  0.09112737 0.355260171 0.237326711 0.3162857465
## 63  0.71640118 0.160099555 0.059794905 0.0637043603
## 64  0.68978855 0.074951936 0.196953651 0.0383058646
## 65  0.24584573 0.263495610 0.306786352 0.1838723053
## 66  0.48634295 0.333526201 0.065033183 0.1150976654
## 67  0.48097184 0.294857057 0.118017180 0.1061539258
## 68  0.28287215 0.470988561 0.015687023 0.2304522657
## 69  0.63518573 0.161312025 0.118827358 0.0846748914
## 70  0.29572515 0.421126351 0.017422902 0.2657255997
## 71  0.59790638 0.190012349 0.060773346 0.1513079270
## 72  0.89895280 0.062362186 0.015788743 0.0228962718
## 73  0.08980060 0.479263627 0.134532274 0.2964034962
## 74  0.45059741 0.306747310 0.088195719 0.1544595606
## 75  0.15500031 0.508761037 0.062206183 0.2740324658
## 76  0.17613406 0.460847014 0.116669967 0.2463489549
## 77  0.78807988 0.137113460 0.013445658 0.0613609992
## 78  0.08118916 0.427952239 0.220075275 0.2707833280
## 79  0.24316407 0.268511232 0.305623897 0.1827007982
## 80  0.19489765 0.460506510 0.086579819 0.2580160217
## 81  0.61662852 0.183408415 0.074993787 0.1249692725
## 82  0.22733623 0.252870706 0.296780832 0.2230122357
## 83  0.35218072 0.359854286 0.085038384 0.2029266066
## 84  0.69474731 0.157104992 0.049863578 0.0982841214
## 85  0.08426726 0.526665662 0.054733611 0.3343334655
## 86  0.17606485 0.442773269 0.046358533 0.3348033438
## 87  0.80600323 0.080039167 0.031426558 0.0825310409
## 88  0.96940036 0.016788742 0.009351367 0.0044595304
## 89  0.23439139 0.487659694 0.090080723 0.1878681935
## 90  0.29032307 0.410204577 0.084029563 0.2154427889
## 91  0.26895505 0.390272548 0.190705773 0.1500666315
## 92  0.61401798 0.141296409 0.161217954 0.0834676532
## 93  0.39713261 0.388079056 0.089522478 0.1252658579
## 94  0.75594579 0.136557671 0.056498650 0.0509978892
## 95  0.17376027 0.457755318 0.091657133 0.2768272764
## 96  0.57845179 0.251532499 0.046386153 0.1236295606
## 97  0.14279474 0.417734513 0.138545409 0.3009253405
## 98  0.06860272 0.420493372 0.188673852 0.3222300596
## 99  0.71045634 0.164647282 0.056558845 0.0683375283
## 100 0.10379192 0.535375053 0.073587970 0.2872450562
##
## $terms
## status ~ spell + age + reprate + disrate + logwage + tenure
## attr(,"variables")
## list(status, spell, age, reprate, disrate, logwage, tenure)
## attr(,"factors")
##         spell age reprate disrate logwage tenure
## status      0   0       0       0       0      0
## spell       1   0       0       0       0      0
## age         0   1       0       0       0      0
## reprate     0   0       1       0       0      0
## disrate     0   0       0       1       0      0
## logwage     0   0       0       0       1      0
```

```
## tenure      0   0       0       0       0      1
## attr(,"term.labels")
## [1] "spell"    "age"     "reprate" "disrate" "logwage" "tenure"
## attr(,"order")
## [1] 1 1 1 1 1 1
## attr(,"intercept")
## [1] 1
## attr(,"response")
## [1] 1
## attr(,".Environment")
## <environment: R_GlobalEnv>
## attr(,"predvars")
## list(status, spell, age, reprate, disrate, logwage, tenure)
## attr(,"dataClasses")
##    status     spell       age   reprate   disrate   logwage    tenure
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##
## $call
## lda(formula = status ~ spell + age + reprate + disrate + logwage +
##     tenure, data = unemp, CV = T)
##
## $xlevels
## named list()
```

```r
unemp_pred <- predict(unemp_lda)
table(unemp$status, unemp_lda_cv$class, dnn = c("actual", "predicted"))
```

```
##       predicted
## actual  0  1  2  3
##      0 17 14  2  1
##      1  6 25  2  1
##      2  4  7  0  0
##      3  5 16  0  0
```

```r
table(unemp$status, unemp_pred$class, dnn = c("actual", "predicted"))
```

```
##       predicted
## actual  0  1  2  3
##      0 21 12  1  0
##      1  6 27  0  1
##      2  4  7  0  0
##      3  5 14  0  2
```

```r
#Second LDA analysis - two groups
unemp2 <- unemp %>%
  mutate(status2 = censor1 + censor2 + censor3)
unemp2_lda <- lda(status2 ~ spell + age + reprate + disrate + logwage + tenure, data = unemp2)
unemp2_lda_cv <- lda(status2 ~ spell + age + reprate + disrate + logwage + tenure,
                     data = unemp2, CV = T)
unemp2_lda
```

```
## Call:
## lda(status2 ~ spell + age + reprate + disrate + logwage + tenure,
##     data = unemp2)
##
## Prior probabilities of groups:
```

```
##    0    1
## 0.34 0.66
##
## Group means:
##      spell      age   reprate   disrate  logwage   tenure
## 0 8.617647 41.35294 0.4627059 0.1042353 5.657338 5.323529
## 1 3.666667 35.19697 0.4748030 0.1113333 5.609623 4.045455
##
## Coefficients of linear discriminants:
##                 LD1
## spell   -0.195994175
## age     -0.044191313
## reprate -0.638542322
## disrate  0.229578725
## logwage  0.009982737
## tenure   0.019347746
```

```
unemp2_lda_cv
```

```
## $class
##   [1] 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 0 0 1 1
##  [75] 1 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1
## Levels: 0 1
##
## $posterior
##             0          1
## 1   0.13659961 0.86340039
## 2   0.06660794 0.93339206
## 3   0.09899563 0.90100437
## 4   0.74852558 0.25147442
## 5   0.05529857 0.94470143
## 6   0.25309691 0.74690309
## 7   0.25225364 0.74774636
## 8   0.28977675 0.71022325
## 9   0.18871086 0.81128914
## 10  0.35027557 0.64972443
## 11  0.18272145 0.81727855
## 12  0.09687253 0.90312747
## 13  0.14300514 0.85699486
## 14  0.16508105 0.83491895
## 15  0.17801320 0.82198680
## 16  0.13708631 0.86291369
## 17  0.17292020 0.82707980
## 18  0.43467336 0.56532664
## 19  0.07740170 0.92259830
## 20  0.96766918 0.03233082
## 21  0.17514135 0.82485865
## 22  0.10531592 0.89468408
## 23  0.18764948 0.81235052
## 24  0.13989106 0.86010894
## 25  0.33335746 0.66664254
## 26  0.21772181 0.78227819
## 27  0.71305052 0.28694948
## 28  0.09601527 0.90398473
```

```
## 29  0.14998162 0.85001838
## 30  0.29061095 0.70938905
## 31  0.20411377 0.79588623
## 32  0.43114819 0.56885181
## 33  0.99457093 0.00542907
## 34  0.54099297 0.45900703
## 35  0.36269967 0.63730033
## 36  0.17967161 0.82032839
## 37  0.07779258 0.92220742
## 38  0.26418639 0.73581361
## 39  0.48745035 0.51254965
## 40  0.23188769 0.76811231
## 41  0.16217973 0.83782027
## 42  0.15187915 0.84812085
## 43  0.11771219 0.88228781
## 44  0.21612283 0.78387717
## 45  0.19212765 0.80787235
## 46  0.14342612 0.85657388
## 47  0.09137465 0.90862535
## 48  0.25665462 0.74334538
## 49  0.71705078 0.28294922
## 50  0.15629650 0.84370350
## 51  0.08307849 0.91692151
## 52  0.41573435 0.58426565
## 53  0.11146107 0.88853893
## 54  0.10653120 0.89346880
## 55  0.37445607 0.62554393
## 56  0.09386292 0.90613708
## 57  0.26462461 0.73537539
## 58  0.13217949 0.86782051
## 59  0.08707094 0.91292906
## 60  0.48904211 0.51095789
## 61  0.11850746 0.88149254
## 62  0.08590867 0.91409133
## 63  0.71304078 0.28695922
## 64  0.76775316 0.23224684
## 65  0.25115166 0.74884834
## 66  0.48086928 0.51913072
## 67  0.47402542 0.52597458
## 68  0.32574152 0.67425848
## 69  0.63559465 0.36440535
## 70  0.33535226 0.66464774
## 71  0.59352347 0.40647653
## 72  0.90059715 0.09940285
## 73  0.08280379 0.91719621
## 74  0.43904302 0.56095698
## 75  0.15441395 0.84558605
## 76  0.16538759 0.83461241
## 77  0.80110182 0.19889818
## 78  0.07451312 0.92548688
## 79  0.25138160 0.74861840
## 80  0.18775376 0.81224624
## 81  0.60908658 0.39091342
## 82  0.23160460 0.76839540
```

```
## 83  0.34226108 0.65773892
## 84  0.68911029 0.31088971
## 85  0.08312669 0.91687331
## 86  0.18236580 0.81763420
## 87  0.81010547 0.18989453
## 88  0.97234544 0.02765456
## 89  0.22634840 0.77365160
## 90  0.28216030 0.71783970
## 91  0.25782190 0.74217810
## 92  0.64488714 0.35511286
## 93  0.39164135 0.60835865
## 94  0.75511749 0.24488251
## 95  0.16679561 0.83320439
## 96  0.57611252 0.42388748
## 97  0.13370688 0.86629312
## 98  0.06336164 0.93663836
## 99  0.70387180 0.29612820
## 100 0.10081157 0.89918843
##
## $terms
## status2 ~ spell + age + reprate + disrate + logwage + tenure
## attr(,"variables")
## list(status2, spell, age, reprate, disrate, logwage, tenure)
## attr(,"factors")
##         spell age reprate disrate logwage tenure
## status2     0   0       0       0       0      0
## spell       1   0       0       0       0      0
## age         0   1       0       0       0      0
## reprate     0   0       1       0       0      0
## disrate     0   0       0       1       0      0
## logwage     0   0       0       0       1      0
## tenure      0   0       0       0       0      1
## attr(,"term.labels")
## [1] "spell"   "age"     "reprate" "disrate" "logwage" "tenure"
## attr(,"order")
## [1] 1 1 1 1 1 1
## attr(,"intercept")
## [1] 1
## attr(,"response")
## [1] 1
## attr(,".Environment")
## <environment: R_GlobalEnv>
## attr(,"predvars")
## list(status2, spell, age, reprate, disrate, logwage, tenure)
## attr(,"dataClasses")
##   status2     spell       age   reprate   disrate   logwage    tenure
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##
## $call
## lda(formula = status2 ~ spell + age + reprate + disrate + logwage +
##     tenure, data = unemp2, CV = T)
##
## $xlevels
## named list()
```

```r
unemp2_pred <- predict(unemp2_lda)
table(unemp2$status2, unemp2_pred$class, dnn = c('actual', 'predicted'))
```

```
##       predicted
## actual  0  1
##      0 14 20
##      1  6 60
```

```r
table(unemp2$status2, unemp2_lda_cv$class, dnn = c('actual', 'predicted'))
```

```
##       predicted
## actual  0  1
##      0 13 21
##      1  7 59
```

```r
#Third LDA analysis - unemployment insurance
unemp <- unemp %>%
  mutate(ui = ifelse(ui == 'yes', 1, 0))
unemp_lda_ui <- lda(ui ~ spell + age + reprate + disrate + logwage + tenure, data = unemp)
unemp_lda_ui_cv <- lda(ui ~ spell + age + reprate + disrate + logwage + tenure,
                       data = unemp, CV = T)
unemp_lda_ui
```

```
## Call:
## lda(ui ~ spell + age + reprate + disrate + logwage + tenure,
##     data = unemp)
##
## Prior probabilities of groups:
##    0    1
## 0.43 0.57
##
## Group means:
##      spell      age   reprate   disrate  logwage   tenure
## 0 3.790698 33.90698 0.4789302 0.1182093 5.566305 3.139535
## 1 6.526316 39.84211 0.4644737 0.1019123 5.670763 5.491228
##
## Coefficients of linear discriminants:
##                 LD1
## spell    0.13807182
## age      0.04479743
## reprate  2.13086437
## disrate -3.56138941
## logwage  0.29691040
## tenure   0.02872657
```

```r
unemp_lda_ui_cv
```

```
## $class
##   [1] 0 0 0 1 0 1 1 1 0 1 0 0 0 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 1 1 0 1 1 1 1 0 0
##  [38] 1 1 1 1 0 1 0 1 0 0 1 1 0 0 1 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1 1 1 1 1 1 0 1
##  [75] 1 0 1 0 1 1 1 1 0 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 0
## Levels: 0 1
##
## $posterior
##             0         1
## 1   0.51566181 0.4843382
```

```
## 2   0.66818095 0.3318190
## 3   0.61172459 0.3882754
## 4   0.29853353 0.7014665
## 5   0.60205466 0.3979453
## 6   0.40433990 0.5956601
## 7   0.41172851 0.5882715
## 8   0.37178173 0.6282183
## 9   0.60167917 0.3983208
## 10  0.39278907 0.6072109
## 11  0.58779193 0.4122081
## 12  0.64135807 0.3586419
## 13  0.60105048 0.3989495
## 14  0.51808040 0.4819196
## 15  0.49190764 0.5080924
## 16  0.65367520 0.3463248
## 17  0.30509338 0.6949066
## 18  0.34932794 0.6506721
## 19  0.67708948 0.3229105
## 20  0.09977494 0.9002251
## 21  0.55092053 0.4490795
## 22  0.44110908 0.5588909
## 23  0.39152989 0.6084701
## 24  0.62005021 0.3799498
## 25  0.44749191 0.5525081
## 26  0.48127943 0.5187206
## 27  0.14697849 0.8530215
## 28  0.58841357 0.4115864
## 29  0.44742682 0.5525732
## 30  0.12442168 0.8755783
## 31  0.53983423 0.4601658
## 32  0.30782160 0.6921784
## 33  0.01709824 0.9829018
## 34  0.29825839 0.7017416
## 35  0.38819002 0.6118100
## 36  0.53244973 0.4675503
## 37  0.67932135 0.3206787
## 38  0.45430297 0.5456970
## 39  0.39181467 0.6081853
## 40  0.46345289 0.5365471
## 41  0.45985758 0.5401424
## 42  0.63394479 0.3660552
## 43  0.43026224 0.5697378
## 44  0.52679475 0.4732052
## 45  0.43392025 0.5660798
## 46  0.55522800 0.4447720
## 47  0.70902898 0.2909710
## 48  0.30306013 0.6969399
## 49  0.28952135 0.7104786
## 50  0.59914353 0.4008565
## 51  0.54459020 0.4554098
## 52  0.44162977 0.5583702
## 53  0.71057182 0.2894282
## 54  0.60246913 0.3975309
## 55  0.47815611 0.5218439
```

```
## 56  0.52063106 0.4793689
## 57  0.48385885 0.5161411
## 58  0.52284714 0.4771529
## 59  0.57658048 0.4234195
## 60  0.30872456 0.6912754
## 61  0.57670193 0.4232981
## 62  0.64187849 0.3581215
## 63  0.26879258 0.7312074
## 64  0.20497103 0.7950290
## 65  0.56843080 0.4315692
## 66  0.23197527 0.7680247
## 67  0.27504206 0.7249579
## 68  0.22447008 0.7755299
## 69  0.28360790 0.7163921
## 70  0.23046063 0.7695394
## 71  0.21805591 0.7819441
## 72  0.15595211 0.8440479
## 73  0.66073570 0.3392643
## 74  0.32071950 0.6792805
## 75  0.49727277 0.5027272
## 76  0.56629904 0.4337010
## 77  0.07516242 0.9248376
## 78  0.67117562 0.3288244
## 79  0.43744193 0.5625581
## 80  0.48406409 0.5159359
## 81  0.18917902 0.8108210
## 82  0.36346733 0.6365327
## 83  0.50014728 0.4998527
## 84  0.30298091 0.6970191
## 85  0.62536974 0.3746303
## 86  0.50913215 0.4908679
## 87  0.08539447 0.9146055
## 88  0.07658238 0.9234176
## 89  0.37211719 0.6278828
## 90  0.29976855 0.7002314
## 91  0.51914532 0.4808547
## 92  0.29336180 0.7066382
## 93  0.32891873 0.6710813
## 94  0.18550111 0.8144989
## 95  0.47522054 0.5247795
## 96  0.19567582 0.8043242
## 97  0.70659227 0.2934077
## 98  0.71168427 0.2883157
## 99  0.16982710 0.8301729
## 100 0.58237154 0.4176285
##
## $terms
## ui ~ spell + age + reprate + disrate + logwage + tenure
## attr(,"variables")
## list(ui, spell, age, reprate, disrate, logwage, tenure)
## attr(,"factors")
##         spell age reprate disrate logwage tenure
## ui          0   0       0       0       0      0
## spell       1   0       0       0       0      0
```

```
## age          0   1       0       0       0      0
## reprate      0   0       1       0       0      0
## disrate      0   0       0       1       0      0
## logwage      0   0       0       0       1      0
## tenure       0   0       0       0       0      1
## attr(,"term.labels")
## [1] "spell"   "age"     "reprate" "disrate" "logwage" "tenure"
## attr(,"order")
## [1] 1 1 1 1 1 1
## attr(,"intercept")
## [1] 1
## attr(,"response")
## [1] 1
## attr(,".Environment")
## <environment: R_GlobalEnv>
## attr(,"predvars")
## list(ui, spell, age, reprate, disrate, logwage, tenure)
## attr(,"dataClasses")
##        ui     spell       age   reprate   disrate   logwage    tenure
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##
## $call
## lda(formula = ui ~ spell + age + reprate + disrate + logwage +
##     tenure, data = unemp, CV = T)
##
## $xlevels
## named list()
```

```
unemp_ui_pred <- predict(unemp_lda_ui)
table(unemp$ui, unemp_ui_pred$class, dnn = c('actual', 'predicted'))
```

```
##       predicted
## actual  0  1
##      0 25 18
##      1 16 41
```

```
table(unemp$ui, unemp_lda_ui_cv$class, dnn = c('actual', 'predicted'))
```

```
##       predicted
## actual  0  1
##      0 22 21
##      1 19 38
```

```
#QDA analysis
unemp2_0 <- unemp2 %>%
  filter(status2 == 0) %>%
  select(spell, age, reprate, disrate, logwage, tenure)
unemp2_1 <- unemp2 %>%
  filter(status2 == 1) %>%
  select(spell, age, reprate, disrate, logwage, tenure)
cov(unemp2_0)
```

```
##               spell          age      reprate      disrate      logwage
## spell   42.78877005   6.10873440 -0.057206774  0.046516934  0.123125134
## age      6.10873440 142.17468806 -0.323680927  0.043368984  2.175352674
## reprate -0.05720677  -0.32368093  0.010356578  0.002255799 -0.038737114
```

```
## disrate  0.04651693   0.04336898  0.002255799  0.005358367 -0.007644404
## logwage  0.12312513   2.17535267 -0.038737114 -0.007644404  0.279157000
## tenure   4.76381462  41.63992870 -0.142477718 -0.047866310  1.370970481
##              tenure
## spell    4.76381462
## age     41.63992870
## reprate -0.14247772
## disrate -0.04786631
## logwage  1.37097048
## tenure  57.55882353
```

```r
cov(unemp2_1)
```

```
##                spell         age      reprate      disrate     logwage
## spell    9.517948718  -0.4256410 -0.003851282 -0.024856410 -0.14870641
## age     -0.425641026 119.1759907 -0.311683683 -0.263528205  1.84501903
## reprate -0.003851282  -0.3116837  0.009236776  0.002755205 -0.02784437
## disrate -0.024856410  -0.2635282  0.002755205  0.004373703 -0.01185040
## logwage -0.148706410   1.8450190 -0.027844375 -0.011850399  0.22266467
## tenure   0.584615385  32.5755245 -0.113160140 -0.030507692  0.52800015
##              tenure
## spell    0.58461538
## age     32.57552448
## reprate -0.11316014
## disrate -0.03050769
## logwage  0.52800015
## tenure  35.55174825
```

```r
unemp2_qda <- qda(status2 ~ spell + age + reprate + disrate + logwage + tenure, data = unemp2)
unemp2_qda_cv <- qda(status2 ~ spell + age + reprate + disrate + logwage + tenure,
                     data = unemp2, CV = T)
unemp2_qda
```

```
## Call:
## qda(status2 ~ spell + age + reprate + disrate + logwage + tenure,
##     data = unemp2)
##
## Prior probabilities of groups:
##    0    1
## 0.34 0.66
##
## Group means:
##      spell      age   reprate   disrate  logwage   tenure
## 0 8.617647 41.35294 0.4627059 0.1042353 5.657338 5.323529
## 1 3.666667 35.19697 0.4748030 0.1113333 5.609623 4.045455
```

```r
unemp2_qda_cv
```

```
## $class
##   [1] 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1
##  [38] 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1
##  [75] 1 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1
## Levels: 0 1
##
## $posterior
##              0            1
```

```
## 1   0.147357506 8.526425e-01
## 2   0.058227393 9.417726e-01
## 3   0.074242429 9.257576e-01
## 4   0.997597082 2.402918e-03
## 5   0.176569606 8.234304e-01
## 6   0.575082576 4.249174e-01
## 7   0.194410875 8.055891e-01
## 8   0.102454471 8.975455e-01
## 9   0.125985773 8.740142e-01
## 10  0.790026171 2.099738e-01
## 11  0.082087905 9.179121e-01
## 12  0.002683024 9.973170e-01
## 13  0.057427072 9.425729e-01
## 14  0.165811227 8.341888e-01
## 15  0.141405440 8.585946e-01
## 16  0.050898799 9.491012e-01
## 17  0.203784178 7.962158e-01
## 18  0.203313898 7.966861e-01
## 19  0.057143372 9.428566e-01
## 20  0.999998861 1.139379e-06
## 21  0.097533503 9.024665e-01
## 22  0.214645400 7.853546e-01
## 23  0.161840172 8.381598e-01
## 24  0.138957364 8.610426e-01
## 25  0.222459769 7.775402e-01
## 26  0.079849566 9.201504e-01
## 27  0.980546839 1.945316e-02
## 28  0.071777753 9.282222e-01
## 29  0.162030604 8.379694e-01
## 30  0.002698050 9.973019e-01
## 31  0.158994089 8.410059e-01
## 32  0.134927352 8.650726e-01
## 33  1.000000000 2.899333e-12
## 34  0.345445948 6.545541e-01
## 35  0.171164013 8.288360e-01
## 36  0.074600623 9.253994e-01
## 37  0.067226641 9.327734e-01
## 38  0.175107073 8.248929e-01
## 39  0.876684643 1.233154e-01
## 40  0.078744800 9.212552e-01
## 41  0.109208486 8.907915e-01
## 42  0.146953515 8.530465e-01
## 43  0.864750806 1.352492e-01
## 44  0.190531073 8.094689e-01
## 45  0.086832661 9.131673e-01
## 46  0.061628788 9.383712e-01
## 47  0.020327724 9.796723e-01
## 48  0.080956793 9.190432e-01
## 49  0.977713384 2.228662e-02
## 50  0.072786906 9.272131e-01
## 51  0.205194768 7.948052e-01
## 52  0.317972942 6.820271e-01
## 53  0.027028467 9.729715e-01
## 54  0.052423061 9.475769e-01
```

```
## 55  0.192929782 8.070702e-01
## 56  0.107221733 8.927783e-01
## 57  0.177958587 8.220414e-01
## 58  0.090074961 9.099250e-01
## 59  0.062976131 9.370239e-01
## 60  0.253817480 7.461825e-01
## 61  0.042523007 9.574770e-01
## 62  0.261582063 7.384179e-01
## 63  0.898296704 1.017033e-01
## 64  0.996567589 3.432411e-03
## 65  0.087406919 9.125931e-01
## 66  0.674607664 3.253923e-01
## 67  0.105365187 8.946348e-01
## 68  0.999994836 5.163514e-06
## 69  0.710018033 2.899820e-01
## 70  0.357896301 6.421037e-01
## 71  0.968747396 3.125260e-02
## 72  0.999969964 3.003585e-05
## 73  0.050702290 9.492977e-01
## 74  0.289903452 7.100965e-01
## 75  0.092715288 9.072847e-01
## 76  0.111394202 8.886058e-01
## 77  0.865157673 1.348423e-01
## 78  0.121894745 8.781053e-01
## 79  0.052816820 9.471832e-01
## 80  0.088082865 9.119171e-01
## 81  0.700544790 2.994552e-01
## 82  0.445465198 5.545348e-01
## 83  0.271254171 7.287458e-01
## 84  0.994775730 5.224270e-03
## 85  0.050996125 9.490039e-01
## 86  0.139589416 8.604106e-01
## 87  0.975745293 2.425471e-02
## 88  0.999994980 5.019966e-06
## 89  0.125847491 8.741525e-01
## 90  0.201492854 7.985071e-01
## 91  0.297930457 7.020695e-01
## 92  0.982020452 1.797955e-02
## 93  0.183558965 8.164410e-01
## 94  0.829985440 1.700146e-01
## 95  0.083557687 9.164423e-01
## 96  0.569578783 4.304212e-01
## 97  0.089451670 9.105483e-01
## 98  0.077645802 9.223542e-01
## 99  0.684508060 3.154919e-01
## 100 0.061923977 9.380760e-01
##
## $terms
## status2 ~ spell + age + reprate + disrate + logwage + tenure
## attr(,"variables")
## list(status2, spell, age, reprate, disrate, logwage, tenure)
## attr(,"factors")
##          spell age reprate disrate logwage tenure
## status2      0   0       0       0       0      0
```

```
## spell       1   0       0       0       0      0
## age         0   1       0       0       0      0
## reprate     0   0       1       0       0      0
## disrate     0   0       0       1       0      0
## logwage     0   0       0       0       1      0
## tenure      0   0       0       0       0      1
## attr(,"term.labels")
## [1] "spell"   "age"     "reprate" "disrate" "logwage" "tenure"
## attr(,"order")
## [1] 1 1 1 1 1 1
## attr(,"intercept")
## [1] 1
## attr(,"response")
## [1] 1
## attr(,".Environment")
## <environment: R_GlobalEnv>
## attr(,"predvars")
## list(status2, spell, age, reprate, disrate, logwage, tenure)
## attr(,"dataClasses")
##   status2     spell       age   reprate   disrate   logwage    tenure
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##
## $call
## qda(formula = status2 ~ spell + age + reprate + disrate + logwage +
##     tenure, data = unemp2, CV = T)
##
## $xlevels
## named list()
```

```r
unemp2_qda_pred <- predict(unemp2_qda)
table(unemp2$status2, unemp2_qda_cv$class, dnn = c('actual', 'predicted'))
```

```
##       predicted
## actual  0  1
##      0 14 20
##      1 11 55
```

```r
table(unemp2$status2, unemp2_qda_pred$class, dnn = c('actual', 'predicted'))
```

```
##       predicted
## actual  0  1
##      0 17 17
##      1  7 59
```