S&DS 363

Homework #4: Discriminant Analysis

Liana Wang, Ryo Tamaki, Yavuz Ramiz Colak, David Lieberman

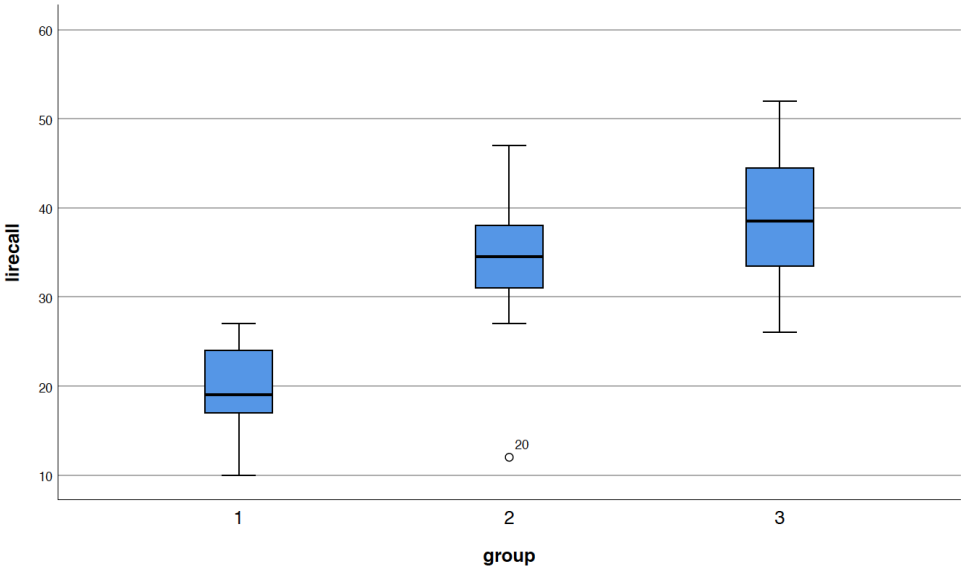
1. Evaluate the assumptions implicit to Discriminant Analysis for your data

i. Evaluating Univariate Normality (Exploratory)

To evaluate univariate normality, we use SPSS to create boxplots of variables of interest by groups.

a. Boxplot of lirecall by group

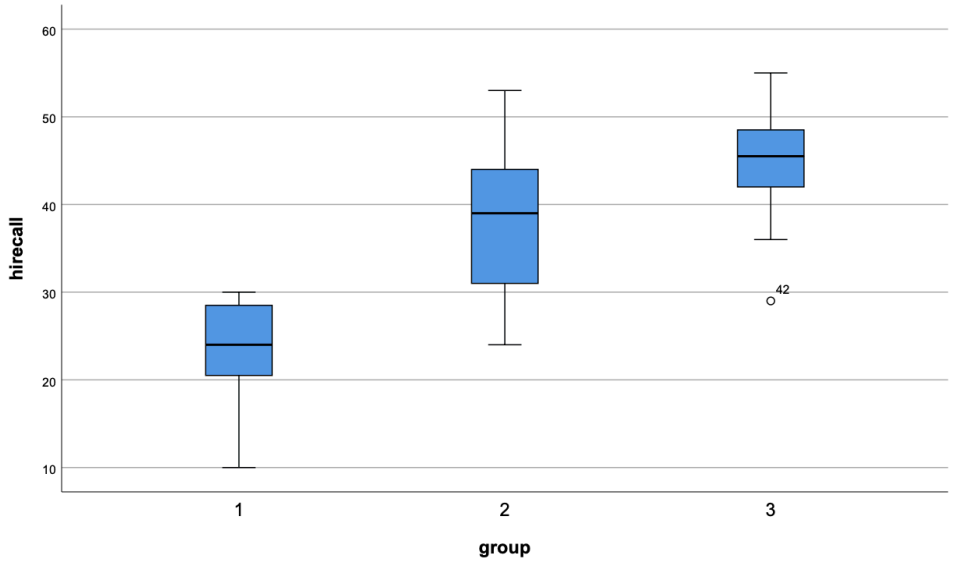
```
EXAMINE VARIABLES=lirecall BY group
  /PLOT=BOXPLOT
  /STATISTICS=NONE
  /NOTOTAL.
```



Our box plot of 'lirecall' shows an approximately normal distribution for each group, although groups 1 and 2 are somewhat skewed left and right, respectively.

b. Boxplot of hirecall by group

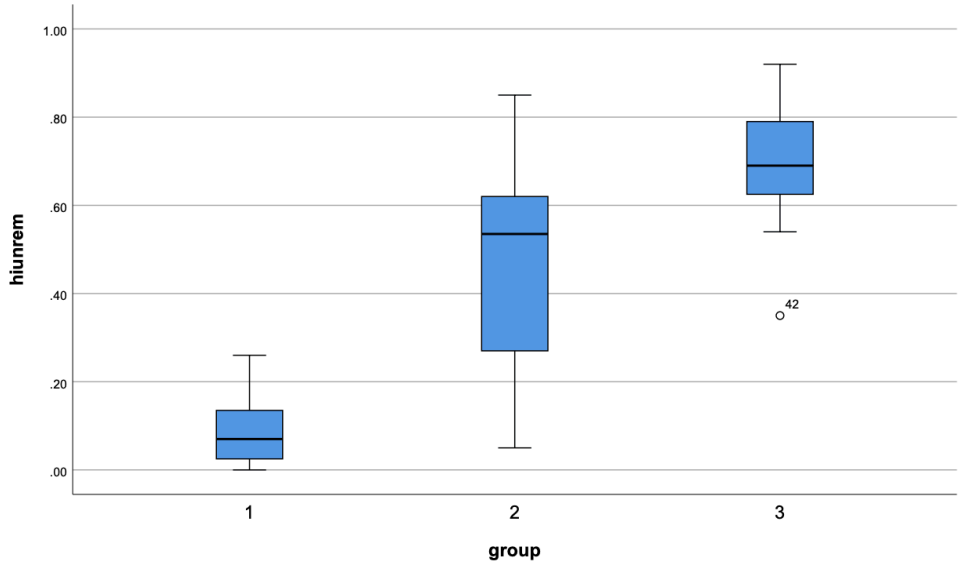
```
EXAMINE VARIABLES=hiunrem BY group
  /PLOT=BOXPLOT
  /STATISTICS=NONE
  /NOTOTAL.
```



Our boxplots of 'hirecall' show an approximately normal distribution for groups 2 and 3 group, while group 1 has a heavy left skew.

c. Boxplot of hiunrem by group

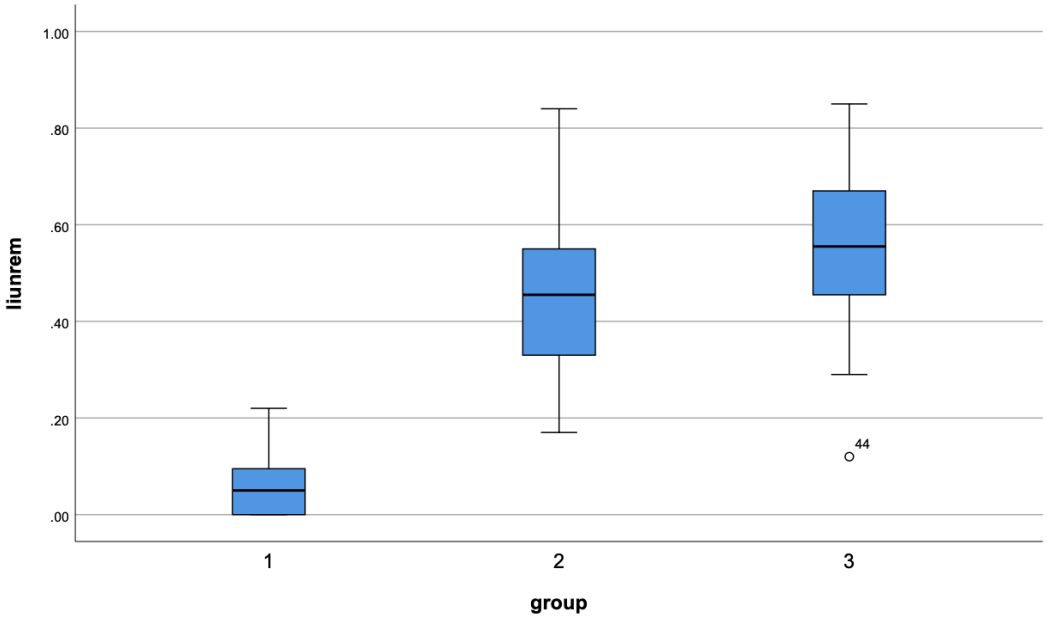
```
EXAMINE VARIABLES=hiunrem BY group
  /PLOT=BOXPLOT
  /STATISTICS=NONE
  /NOTOTAL.
```



The groups seem mostly normally distributed, except group 1 seems to have a moderate right skew this time when measuring HI imagery unreminded memory, as opposed to HI Imagery recall as from our 'hirecall' boxplots in part b.

d. Boxplot of liunrem by group

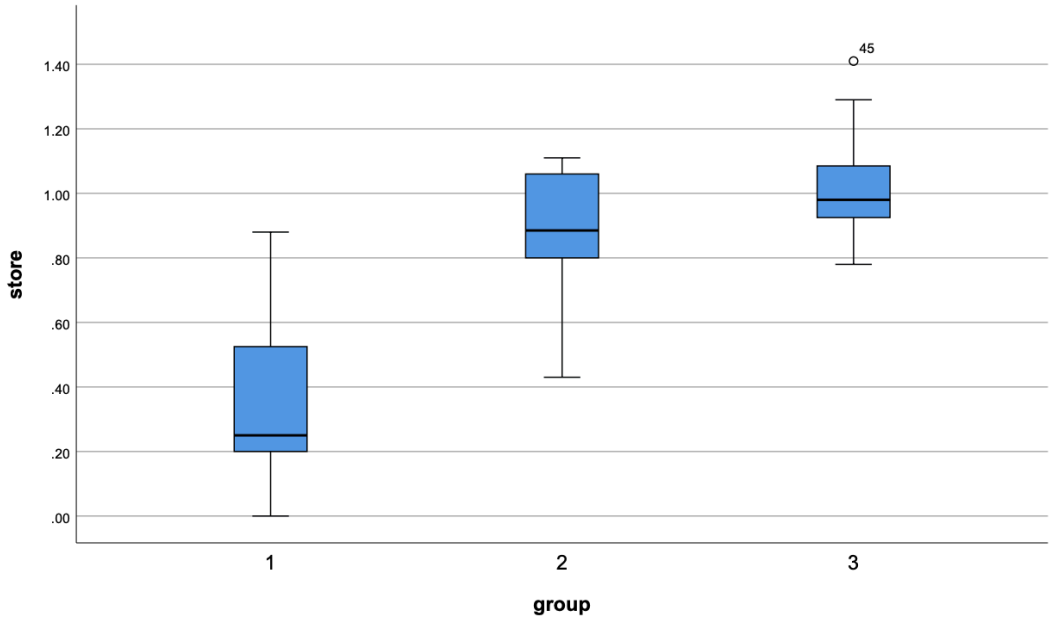
```
EXAMINE VARIABLES=liunrem BY group
  /PLOT=BOXPLOT
  /STATISTICS=NONE
  /NOTOTAL.
```



The groups appear to remain be mostly normally distributed, except for group 1 which has a strong right skew this time when measuring LO imagery unreminded memory, as opposed to LO Imagery recall as from our 'lirecall' boxplots in part a.

e. Boxplot of store by group

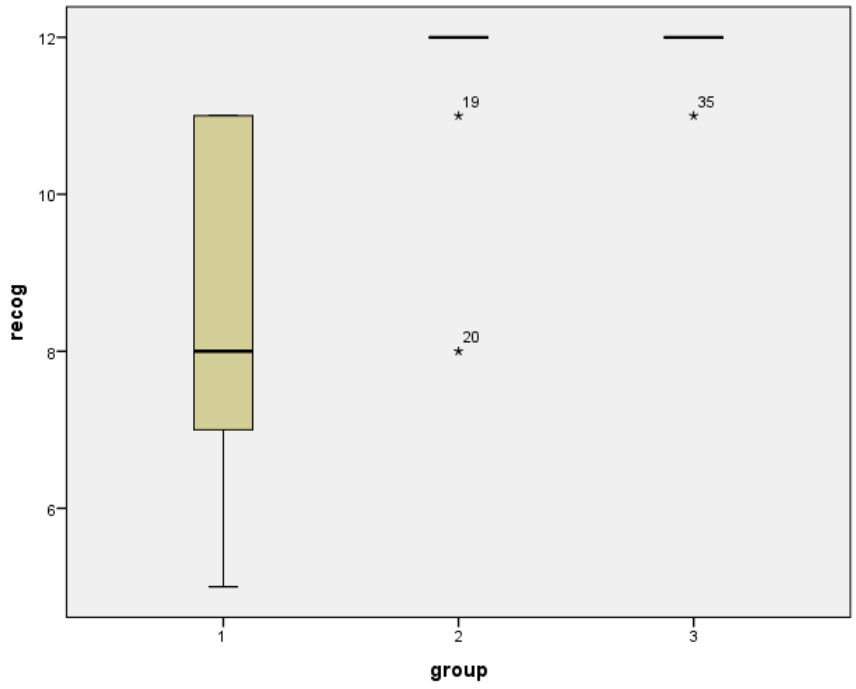
```
EXAMINE VARIABLES=store BY group
  /PLOT=BOXPLOT
  /STATISTICS=NONE
  /NOTOTAL.
```



Groups 1 and 2 are visibly, though not extremely skewed right and left, respectively, while group 3 is approximately Normal (save for one outlier).

f. Boxplot of recog by group

```
EXAMINE VARIABLES=recog BY group
  /PLOT=BOXPLOT
  /STATISTICS=NONE
  /NOTOTAL.
```



This boxplot is very interesting, because it illustrates group 2 and 3's nearly nonexistent spreads. Conversely, group 1, although skewed from Normal, has a very wide range of response values for recognition memory. Note that this is a categorical variable. Regardless, we obtain useful information from these boxplots.

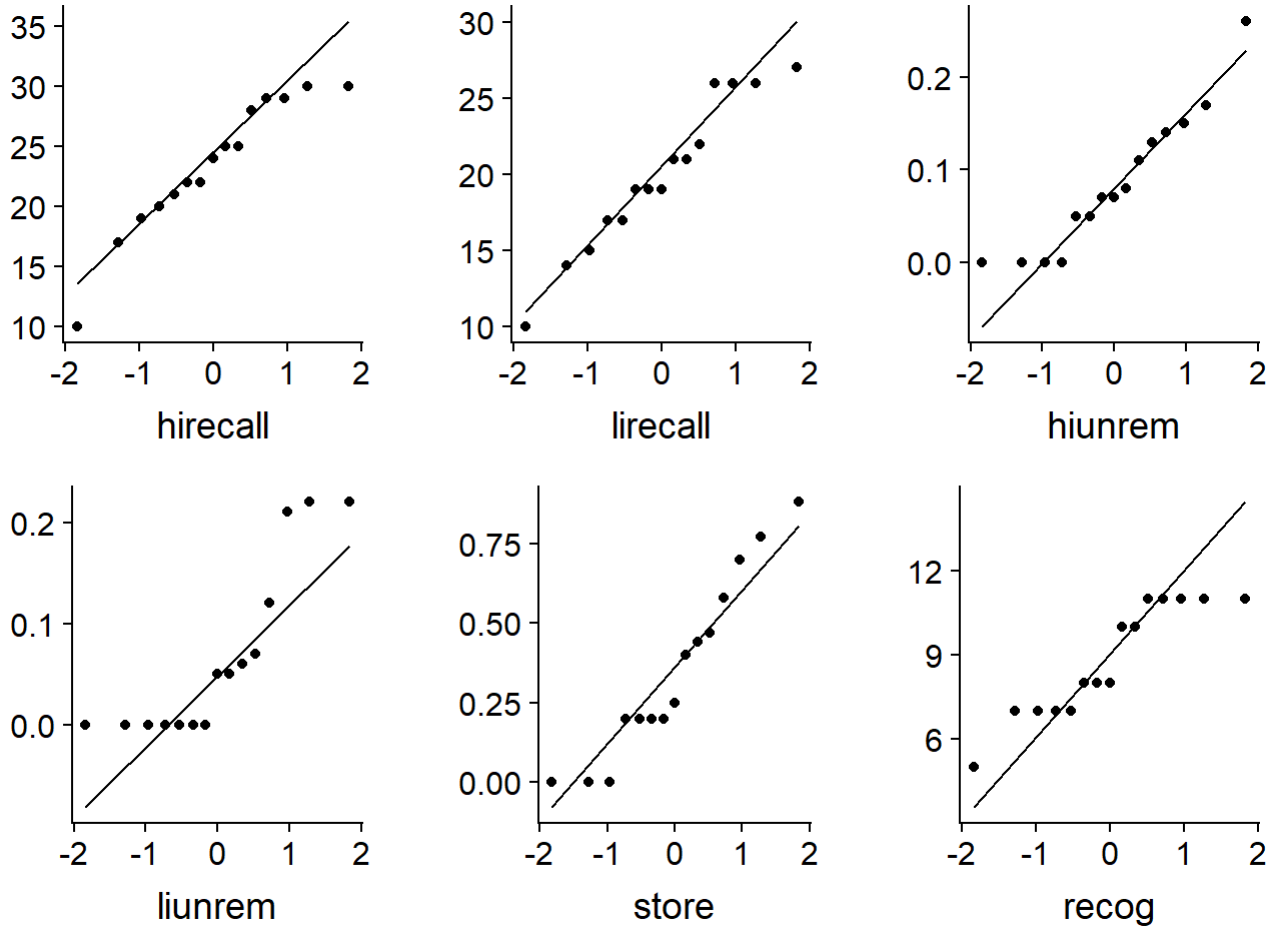
Overall, looking at all these boxplots for each variable, while there are some cases of a group skewed from Normal, the on the whole, each of these variable data seem to approximately follow a univariate Normal distribution.

Addendum: Briefly, we wanted to show that QQ plots show that our data set is approximately normal. Here are the plots from R:

```
QQPlot <- function(x, na.rm = TRUE){
  plots <- list()
  j <- 1
  for (i in names(x)) {
    plots[[i]] <- ggplot(x, aes_string(sample = i)) + stat_qq() + stat_qq_line() + xlab(names(x)
[j]) + ylab("")
    j <- j+1
  }
  plot_grid(plotlist = plots)
}

data_group1 <- data[data$group == 1,c(3:4,6:9)]
data_group2 <- data[data$group == 2,c(3:4,6:9)]
data_group3 <- data[data$group == 3,c(3:4,6:9)]
```

Group 1

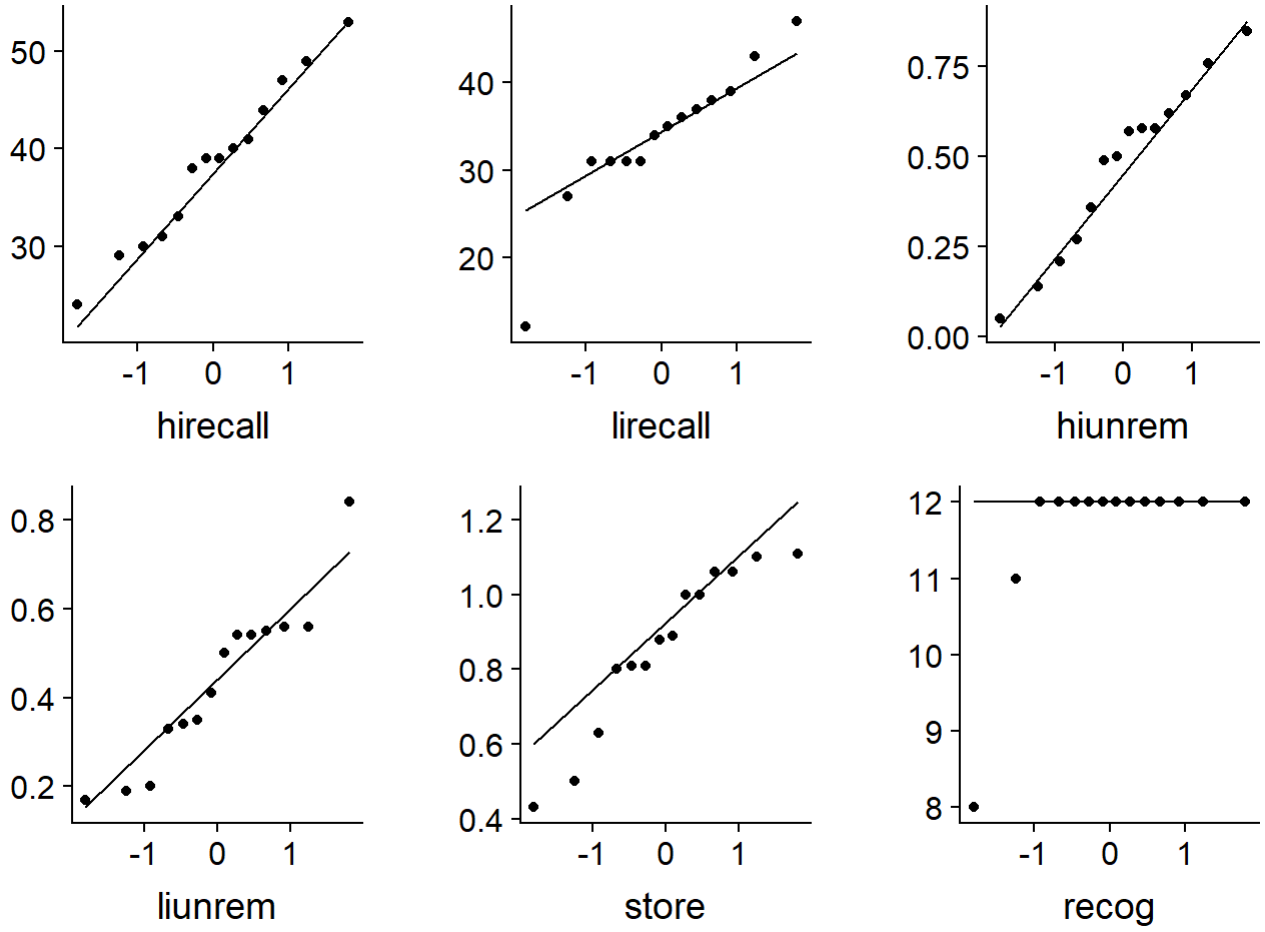
```
QQPlot(data_group1)
```



As observed in the skewness in some of our boxplots of group 1 above, there are some notable deviations from univariate Normality, but on the whole, we don't think these deviations will be serious enough to impede our analysis going forward, and we will conclude that overall that the group 1 data for each of the variables approximately follows a univariate Normal distribution.

Group 2

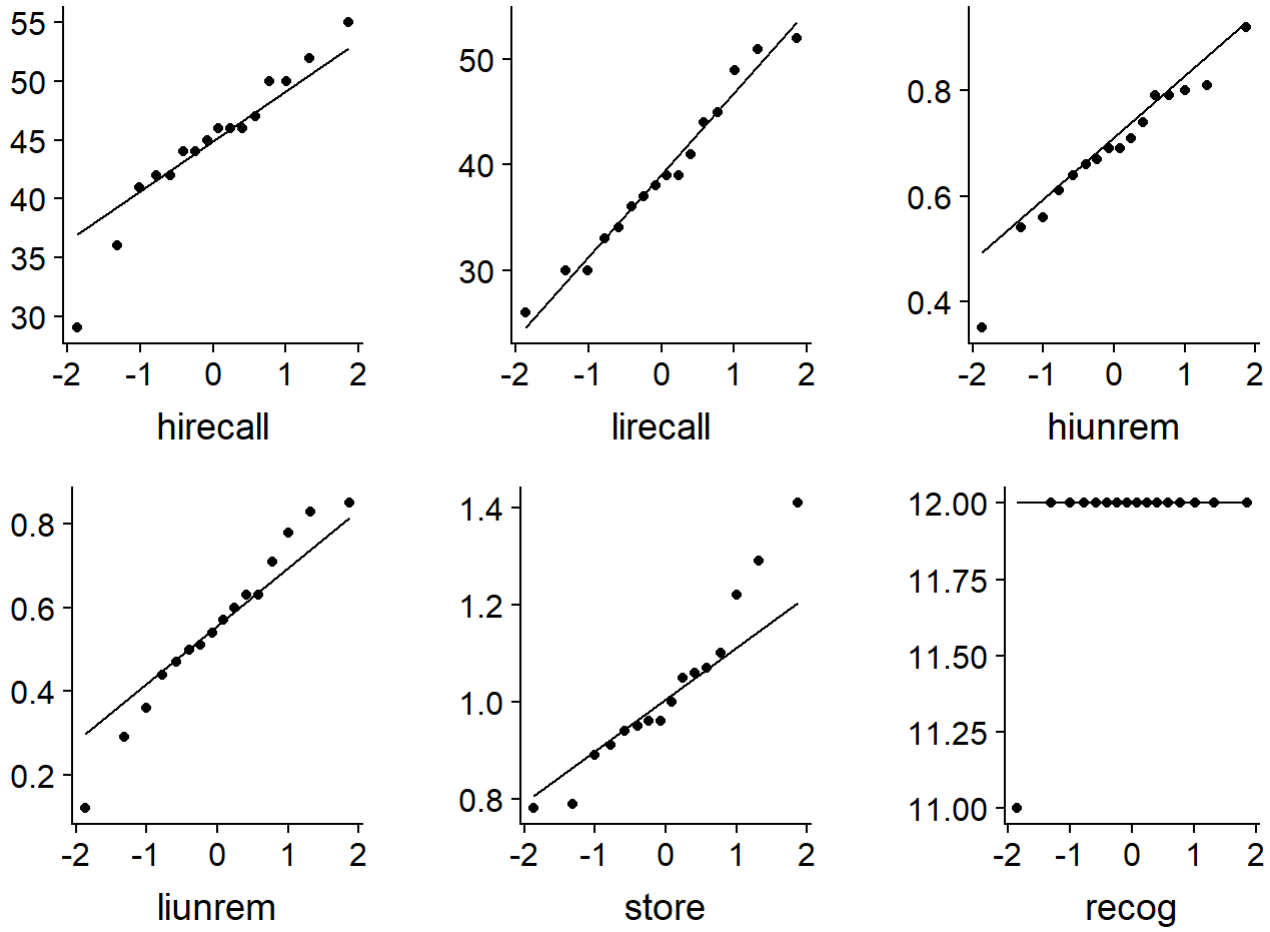
```
QQPlot(data_group2)
```



These univariate quantile-quantile plots look much better for group 2, and despite some deviations for store and liunrem, and single outliers in lirecall and recog, group 2 data for all variables follow an approximately univariate Normal distribution.

Group 3

```
QQPlot(data_group3)
```

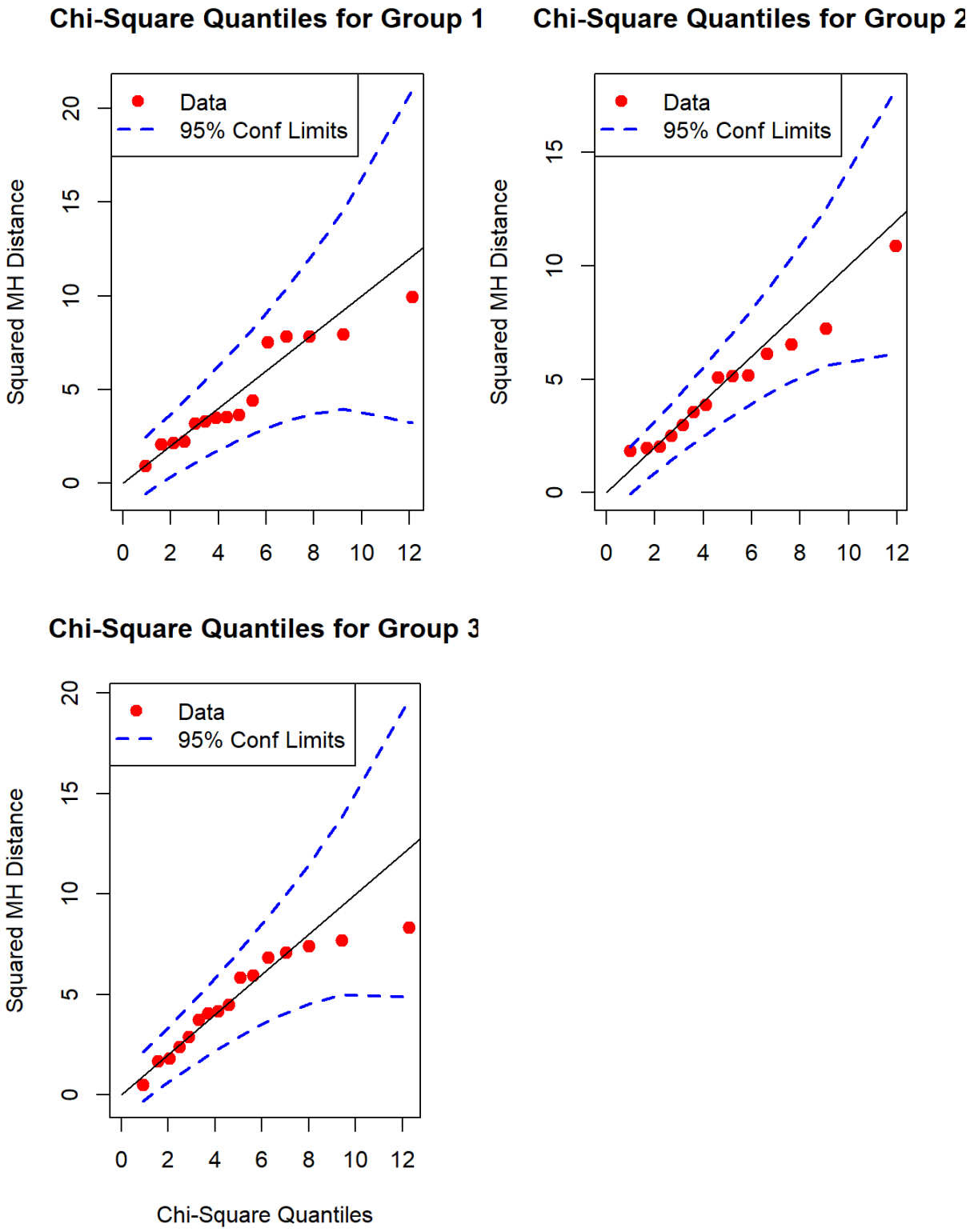


These univariate quantile-quantile plots for group 3 look great! Group 3 data for all variables follow an approximately univariate Normal distribution.

ii. Evaluating Multivariate Normality

To evaluate multivariate normality, we plot quantile plots for each group.

```
source("http://www.reuningscherer.net/STAT660/R/CSQPlot.r.txt")

par(mfrow=c(1,2))
CSQPlot(data[data$group == 1, c(3,4,6,7,8)], label="Group 1")
CSQPlot(data[data$group == 2, c(3,4,6,7,8)], label="Group 2")
CSQPlot(data[data$group == 3, c(3,4,6,7,8)], label="Group 3")
```



Since the data falls within the 95% confidence intervals for every group, it is safe to consider the data roughly multivariate normal and thus, we can use our data set for discriminant analysis. Because normality assumptions are roughly fulfilled, we see no need to transform our data.

iii. Evaluating Similarity of Covariances Matrices

a. Evaluating group statistics to evaluate whether the standard deviations for each group are roughly equal

```
DISCRIMINANT
 /GROUPS=group(1 3)
 /VARIABLES=hirecall lirecall hiunrem liunrem store recog
 /ANALYSIS ALL
 /OUTFILE=MODEL('C:\Users\rt462\Desktop\test.xml')
 /SAVE=SCORES
 /METHOD=WILKS
 /FIN=3.84
 /FOUT=2.71
 /PRIORS EQUAL
 /HISTORY
 /STATISTICS=MEAN STDDEV BOXM TABLE
 /CLASSIFY=NONMISSING SEPARATE.
```

Group Statistics

group		Mean	Std. Deviation	Valid N (listwise) Unweighted	Weighted
1	hirecall	23.4000	5.59081	15	15.000
	lirecall	19.9333	4.96368	15	15.000
	hiunrem	.0853	.07539	15	15.000
	liunrem	.0667	.08541	15	15.000
	store	.3527	.28349	15	15.000
	recog	8.8000	2.00713	15	15.000
2	hirecall	38.3571	8.27979	14	14.000
	lirecall	33.7143	8.19407	14	14.000
	hiunrem	.4750	.23715	14	14.000
	liunrem	.4343	.18567	14	14.000
	store	.8629	.21777	14	14.000
	recog	11.6429	1.08182	14	14.000
3	hirecall	44.6875	6.20450	16	16.000
	lirecall	39.0000	7.67680	16	16.000
	hiunrem	.6856	.13347	16	16.000
	liunrem	.5519	.19559	16	16.000
	store	1.0238	.17021	16	16.000
	recog	11.9375	.25000	16	16.000
Total	hirecall	35.6222	11.24781	45	45.000
	lirecall	31.0000	10.73016	45	45.000
	hiunrem	.4200	.29883	45	45.000
	liunrem	.3536	.26454	45	45.000
	store	.7500	.36685	45	45.000
	recog	10.8000	1.92590	45	45.000

By visual inspection, we can see that the standard deviations between the groups on the same variables are not very similar: for example, the standard deviation on lire call for group 2 is almost twice the standard deviation for group 1, and the standard deviation for recog in group 1 is eight times the standard deviation in group 3.

b. Evaluating log determinants

Log Determinants

group	Rank	Log Determinant
1	2	-4.219
2	2	-2.980
3	2	-6.812
Pooled within-groups	2	-3.222

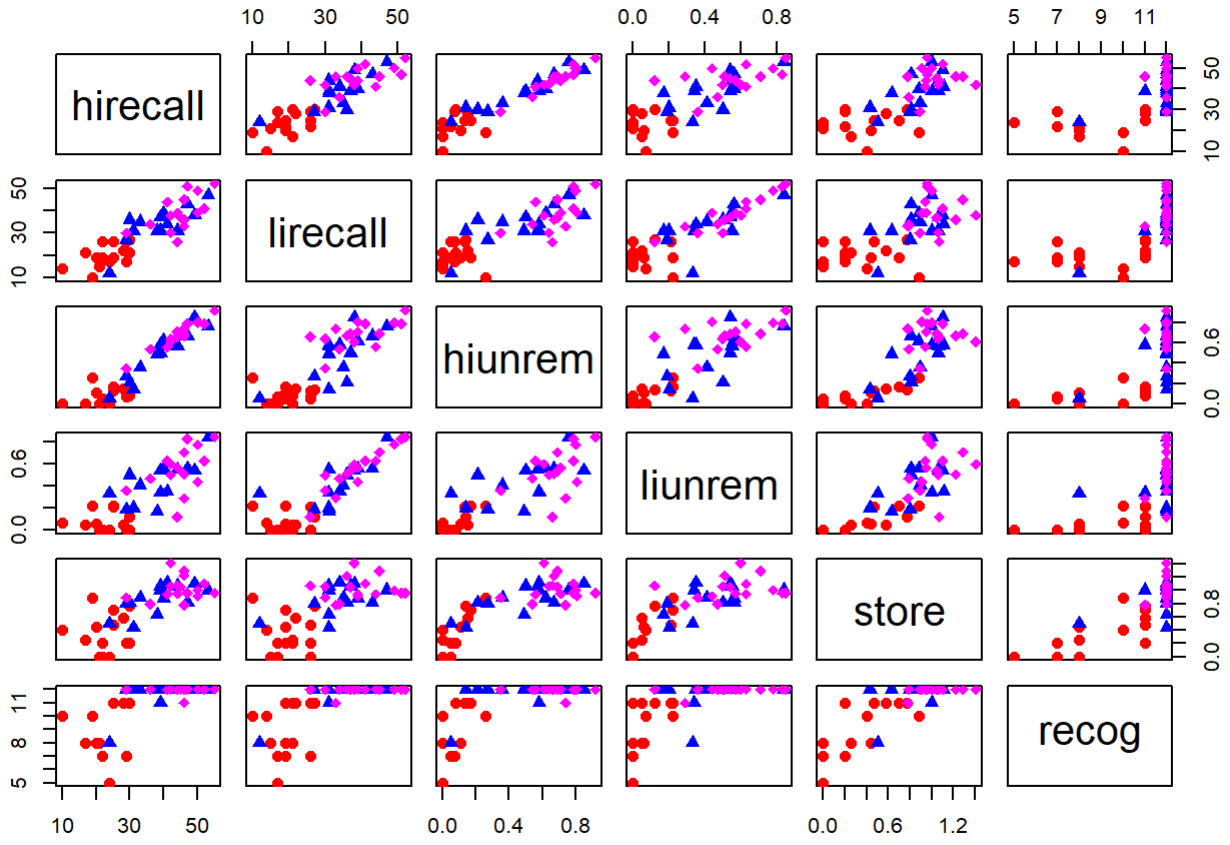
The ranks and natural logarithms of determinants printed are those of the group covariance matrices.

Similarly, the different log determinant figures across the groups suggest that the covariance matrices are unequal.

c. Plotting the Covariance Matrices (For this one, we are using R)

Matrix plot to look at differences between groups. Initial impressions are that the multivariate footprints appear different.

```
plot(data[,c(3:4,6:9)], col = 2*(as.numeric(data$group)), pch = as.numeric(data$group)+15,cex=1.2)
```



```
## [1] "Covariance Matrix for Group 1"
##             hirecall    lirecall      hiunrem      liunrem       store
## hirecall 31.25714286 15.10000000  0.124857143 -0.051428571  0.04100000
## lirecall 15.10000000 24.63809524 -0.013904762 -0.041666667 -0.12623810
## hiunrem   0.12485714 -0.01390476  0.005683810  0.004819048  0.01819190
## liunrem  -0.05142857 -0.04166667  0.004819048  0.007295238  0.02013810
## store     0.04100000 -0.12623810  0.018191905  0.020138095  0.08036381
## recog     1.22857143  1.05714286  0.090428571  0.112857143  0.42057143
##               recog
## hirecall 1.22857143
## lirecall 1.05714286
## hiunrem  0.09042857
## liunrem  0.11285714
## store    0.42057143
## recog    4.02857143
##
##
## [1] "Covariance Matrix for Group 2"
##           hirecall  lirecall    hiunrem    liunrem      store      recog
## hirecall 68.554945 52.648352 1.86884615 1.07527473 1.18813187 4.36813187
## lirecall 52.648352 67.142857 1.40461538 1.02439560 0.94395604 6.89010989
## hiunrem   1.868846  1.404615 0.05624231 0.02670769 0.03874615 0.12269231
## liunrem   1.075275  1.024396 0.02670769 0.03447253 0.02293297 0.03934066
## store     1.188132  0.943956 0.03874615 0.02293297 0.04742198 0.10109890
## recog     4.368132  6.890110 0.12269231 0.03934066 0.10109890 1.17032967
##
##
## [1] "Covariance Matrix for Group 3"
##            hirecall lirecall       hiunrem    liunrem        store
## hirecall 38.4958333 28.46667  0.8045416667 0.56262500 0.1759166667
## lirecall 28.4666667 58.93333  0.6220000000 1.39200000 0.0920000000
## hiunrem   0.8045417  0.62200  0.0178129167 0.01220208 0.0006708333
## liunrem   0.5626250  1.39200  0.0122020833 0.03825625 0.0071725000
## store     0.1759167  0.09200  0.0006708333 0.00717250 0.0289716667
## recog    -0.0875000  0.40000 -0.0036250000 0.01745833 0.0162500000
##               recog
## hirecall -0.08750000
## lirecall  0.40000000
## hiunrem  -0.00362500
## liunrem   0.01745833
## store     0.01625000
## recog     0.06250000
```

By visual inspection, we can see that the covariance matrices seem to be quite different. Some notible examples include the group 2 hirecall-recog covariance to be 4.368, while in group 3 it is -0.0875; additionally, the group 1 hirecall-lirecall covariance is 15.100, while in group 2 it is 52.648.

d. Referring to Box's Test of Equality of Covariance Matrices

```
DISCRIMINANT
 /GROUPS=group(1 3)
 /VARIABLES=hirecall lirecall hiunrem liunrem store recog
 /ANALYSIS ALL
 /METHOD=WILKS
 /FIN=3.84
 /FOUT=2.71
 /PRIORS EQUAL
 /HISTORY
```

Test Results

Box's M		64.659
F	Approx.	10.031
	df1	6
	df2	40401.965
	Sig.	.000

Tests null hypothesis of equal population covariance matrices.

It comes as no surprise that the Box's M test yields an approximately zero p-value, registering a significant difference between each group's covariance matrix, as alluded to from our earlier visual inspection of substantial differences in covariance matrix elements and log-determinants between groups. These significant differences between each group's covariance matrix means that our data set is not suitable for Linear Discriminant Analysis. Therefore, we will proceed by doing Quadratic Discriminant Analysis instead. Despite the differences between covariance matrices, according to our chi-squared quantile-quantile plots above, our groups have an approximately multivariate Normal distribution within groups, which is enough to make our data suitable for QDA.

2. Perform stepwise discriminant analysis on your data. Comment on which model seems the best. Use quadratic discriminant analysis if appropriate.

We perform stepwise quadratic discriminant analysis. We use QDA as opposed to LDA due to the inequalities in our covariance matrices. As discussed later, the SPSS output shows that two variables, hiunrem and recog, are added in this process. So, the model with those two predictors is the best.

```
DISCRIMINANT
 /GROUPS=group(1 3)
 /VARIABLES=hirecall lirecall hiunrem liunrem store recog
 /ANALYSIS ALL
 /METHOD=WILKS
 /FIN=3.84
 /FOUT=2.71
 /PRIORS EQUAL
 /HISTORY
```

Variables Entered/Removed[a,b,c,d]

		Wilks' Lambda				Exact F			Wilks' ... Exact F
Step	Entered	Statistic	df1	df2	df3	Statistic	df1	df2	Sig.
1	hiunrem	.274	1	2	42.000	55.549	2	42.000	.000
2	recog	.230	2	2	42.000	22.209	4	82.000	.000

At each step, the variable that minimizes the overall Wilks' Lambda is entered.

a. Maximum number of steps is 12.

b. Minimum partial F to enter is 3.84.

c. Maximum partial F to remove is 2.71.

d. F level, tolerance, or VIN insufficient for further computation.

Variables in the Analysis

Step		Tolerance	F to Remove	Wilks' Lambda
1	hiunrem	1.000	55.549	
2	hiunrem	.899	19.057	.445
	recog	.899	3.910	.274

The SPSS output shows that after various iterations, two variables, hiunrem and recog, are kept via the stepwise DA process. In step 0, we see that the variable with largest F-statistic is hiunrem, with a value of 55.549. In Step 1, after accepting hiunrem as our first variable in Step 0, we see that recog has an F-statistic of 3.910, which is still significant in that it is still above the minimum F-statistic of 3.84 (corresponding to an alpha of 0.05) that we set as the threshold for entering the model. Therefore, in Step 1, we accept recog as our second variable. In step 2, we see that there are no more variables with F-statistics above our threshold, with the largest being liunrem with 1.295, far below our 3.84 minimum.

As a result, in our best model, we include hiunrem and recog variables.

Variables Not in the Analysis

Step		Tolerance	Min. Tolerance	F to Enter	Wilks' Lambda
0	hirecall	1.000	1.000	40.324	.342
	lirecall	1.000	1.000	29.617	.415
	hiunrem	1.000	1.000	55.549	.274
	liunrem	1.000	1.000	36.524	.365
	store	1.000	1.000	36.144	.367
	recog	1.000	1.000	26.238	.445
1	hirecall	.293	.293	.454	.268
	lirecall	.669	.669	.788	.264
	liunrem	.705	.705	2.130	.249
	store	.748	.748	2.441	.245
	recog	.899	.899	3.910	.230
2	hirecall	.287	.269	.711	.222
	lirecall	.658	.643	.340	.227
	liunrem	.697	.672	1.295	.216
	store	.539	.539	.431	.226

3. Comment of whether there is statistical evidence that the multivariate group means are different.

Tests of Equality of Group Means

	Wilks' Lambda	F	df1	df2	Sig.
hirecall	.342	40.324	2	42	.000
lirecall	.415	29.617	2	42	.000
hiunrem	.274	55.549	2	42	.000
liunrem	.365	36.524	2	42	.000
store	.367	36.144	2	42	.000
recog	.445	26.238	2	42	.000

In considering the results of the test of equality of group means (obtained via SPSS with the Univariate ANOVA option from DA), we find that because the significance values are all zero, group means are different for all variables.

Wilks' Lambda

Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1 through 2	.230	60.921	4	.000
2	.924	3.295	1	.069

Equivalently, we draw this conclusion that the multivariate means are not the same across the groups based on the significance of the Wilks' Lambda statistic. In this case, we observe that the Wilks' Lambda is significant in SPSS (for the functions 1 through 2), with a p-value of essentially 0.

```
summary.manova(alz.manova,test="Wilks")
```

```
##            Df   Wilks approx F num Df den Df    Pr(>F)
## data$group  1 0.27349   16.824      6     38 2.307e-09 ***
## Residuals  43
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We also calculated Wilks' Lambda in R. Considering the results of the multivariate Wilks' Lambda (0.2735) between multivariate group means(MANOVA), we find that the p-value of the approximated F-statistic (16.824) is approximately zero (ie.significant). So the multivariate mean is significantly different between all three groups.

4. How many discriminant functions are significant?

Because we have 3 groups and 6 variables, we have two discriminant functions (min((G -1), p)). However, our results below show that only the first of the two discriminant functions is significant. The p-value of the approximate chi-square test statistic (approximated from the calculated multivariate Wilks' Lambda) is approximately 0 for discriminant functions 1-2, while it is 0.069 (ie. not significant) for the second discriminant function alone. Therefore, we conclude that only the first discriminant function remains significant. Additionally, our posthoc tests corroborate the Wilks' Lambda results. Looking at the significance of the mean difference result in our posthoc tests below, the first function is able to effectively discriminate between the groups such that the mean differences are all significant (the significance probabilities are all below 0.05). Conversely, this is not true for the second discriminant function -- the p-values for the mean differences between groups are all insignificant (the significance probabilities are all above 0.05). It also stands to reason that not all variables should be included in the discriminant functions as several variables are highly correlated, as shown by the sample correlations in the correlations table below. If all variables are included, not all variables will be good discriminators since highly correlated discriminator variables will tend to predict one another.

Wilks' Lambda

Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1 through 2	.230	60.921	4	.000
2	.924	3.295	1	.069

```
CORRELATIONS
 /VARIABLES=hirecall lirecall hiunrem liunrem store recog
 /PRINT=TWOTAIL NOSIG
 /MISSING=PAIRWISE.
```

Correlations

		hirecall	lirecall	hiunrem	liunrem	store	recog
hirecall	Pearson Correlation	1	.869**	.947**	.811**	.746**	.668**
	Sig. (2-tailed)		.000	.000	.000	.000	.000
	N	45	45	45	45	45	45
lirecall	Pearson Correlation	.869**	1	.844**	.878**	.676**	.682**
	Sig. (2-tailed)	.000		.000	.000	.000	.000
	N	45	45	45	45	45	45
hiunrem	Pearson Correlation	.947**	.844**	1	.846**	.832**	.724**
	Sig. (2-tailed)	.000	.000		.000	.000	.000
	N	45	45	45	45	45	45
liunrem	Pearson Correlation	.811**	.878**	.846**	1	.795**	.692**
	Sig. (2-tailed)	.000	.000	.000		.000	.000
	N	45	45	45	45	45	45
store	Pearson Correlation	.746**	.676**	.832**	.795**	1	.826**
	Sig. (2-tailed)	.000	.000	.000	.000		.000
	N	45	45	45	45	45	45
recog	Pearson Correlation	.668**	.682**	.724**	.692**	.826**	1
	Sig. (2-tailed)	.000	.000	.000	.000	.000	
	N	45	45	45	45	45	45

**. Correlation is significant at the 0.01 level (2-tailed).

```
DISCRIMINANT
 /GROUPS=group(1 3)
 /VARIABLES=hirecall lirecall hiunrem liunrem store recog
 /ANALYSIS ALL
 /SAVE=SCORES
 /METHOD=WILKS
 /FIN=3.84
 /FOUT=2.71
 /PRIORS EQUAL
 /HISTORY
 /CLASSIFY=NONMISSING SEPARATE.
```

Post Hoc Tests

Multiple Comparisons

Tukey HSD

Dependent Variable	(I) group	(J) group	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Discriminant Scores from Function 1 for Analysis 1	1	2	-2.79338685*	.37161168	.000	-3.6962151	-1.8905586
		3	-3.95184738*	.35939764	.000	-4.8250017	-3.0786930
	2	1	2.79338685*	.37161168	.000	1.8905586	3.6962151
		3	-1.15846053*	.36596253	.008	-2.0475642	-.2693569
	3	1	3.95184738*	.35939764	.000	3.0786930	4.8250017
		2	1.15846053*	.36596253	.008	.2693569	2.0475642
Discriminant Scores from Function 2 for Analysis 1	1	2	-.51479503	.37161168	.358	-1.4176233	.3880332
		3	.13940919	.35939764	.921	-.7337451	1.0125635
	2	1	.51479503	.37161168	.358	-.3880332	1.4176233
		3	.65420422	.36596253	.186	-.2348995	1.5433079
	3	1	-.13940919	.35939764	.921	-1.0125635	.7337451
		2	-.65420422	.36596253	.186	-1.5433079	.2348995

5. Use classification, both regular and leave-one-out to evaluate the discriminating ability of your functions.

Regular: Quadratic Discriminant Analysis

```
DISCRIMINANT
 /GROUPS=group(1 3)
 /VARIABLES=hirecall lirecall hiunrem liunrem store recog
 /ANALYSIS ALL
 /SAVE=SCORES
 /METHOD=WILKS
 /FIN=3.84
 /FOUT=2.71
 /PRIORS EQUAL
 /HISTORY
 /STATISTICS=TABLE
 /CLASSIFY=NONMISSING SEPARATE.
```

Classification Results[a]

			Predicted Group Membership			
		group	1	2	3	Total
Original	Count	1	14	1	0	15
		2	3	3	8	14
		3	0	2	14	16
	%	1	93.3	6.7	.0	100.0
		2	21.4	21.4	57.1	100.0
		3	.0	12.5	87.5	100.0

a. 68.9% of original grouped cases correctly classified.

Our classification results for quadratic discriminant analysis show that the discriminating ability of our functions were either highly effective or ineffective depending on the group. Group 1 predictions were largely successful. Our function correctly predicted 14/15 Group 1 individuals, with only 1/15 Group 1 individuals being incorrectly placed into Group 2. Group 2 predictions were largely unsuccessful. The discriminant function placed only 3/14 Group 2 individuals correctly, a meager 21.4% success rate, with 78.5% of Group 2 individuals getting placed incorrectly, with the majority incorrectly predicted as being in Group 3. Group 3 predictions were largely successful. Our function correctly predicted 14/16 of Group 3 individuals, while the remaining 12.5% were incorrectly placed into Group 2. Contextualizing these results, it appears that the model is able to discern, with a 93.3% 'accuracy' rate, individuals with Alzheimers from both the control group and individuals with depression. However, the model largely fails at discerning depressed individuals from those who are not, with most truly depressed individuals being categorized as part of the control group. Finally, the model seems to be able to discern individuals in the control group reasonably well with an 87.5% 'accuracy' rate, but again, most of the error seems to arise from differentiating between the control group and those with depression.

Leave-one-out: Linear Discriminant Analysis* We note this is pro-forma

```
DISCRIMINANT
 /GROUPS=group(1 3)
 /VARIABLES=hirecall lirecall hiunrem liunrem store recog
 /ANALYSIS ALL
 /SAVE=SCORES
 /METHOD=WILKS
 /FOUT=2.71
 /PRIORS EQUAL
 /HISTORY
 /STATISTICS=TABLE CROSSVALID
 /CLASSIFY=NONMISSING POOLED.
```

Classification Results[a,c]

			Predicted Group Membership			
		group	1	2	3	Total
Original	Count	1	15	0	0	15
		2	1	8	5	14
		3	0	3	13	16
	%	1	100.0	.0	.0	100.0
		2	7.1	57.1	35.7	100.0
		3	.0	18.8	81.3	100.0
Cross-validated[b]	Count	1	15	0	0	15
		2	2	7	5	14
		3	0	3	13	16
	%	1	100.0	.0	.0	100.0
		2	14.3	50.0	35.7	100.0
		3	.0	18.8	81.3	100.0

a. 80.0% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 77.8% of cross-validated grouped cases correctly classified.

We performed leave-one-out analysis as an exploratory analysis, with the understanding that it usually only works well for linear discriminant analysis, for which our data does not fit the prerequisite assumptions (recall that the covariance matrices are unlike). That said, it appears that in running the leave-one-out analysis using LDA, we find that 80% of the original grouped cases were correctly classified.

Addendum to our solution: Because we cannot include cross validation in SPSS, we also calculated QDA in R for consistency and comprehensiveness.

QDA, no cross validation

```
alz.qda <- qda(data[,c("hirecall","lirecall","hiunrem","liunrem","store","recog")], grouping=dat
a$group, CV=FALSE)
table(data$group, predict(alz.qda)$class)
```

```
##
##      1  2  3
##   1 14  1  0
##   2  0 12  2
##   3  0  0 16
```

When using all the available data to construct the model, our classification results for quadratic discriminant analysis show that the discriminating ability of our model was highly effective. Our model correctly predicted 14/15 group 1 individuals, 12/14 group 2 individuals, and all 16/16 of Group 3 individuals. Contextualizing these results, it appears that the model is able to discern, with a 93.3% 'accuracy' rate, individuals with Alzheimer's from both the control group and individuals with depression. However, the model's biggest failure was in discerning depressed individuals from those who are not, with 14.3% truly depressed individuals being categorized as part of the control group. Overall, when using all the available data to construct the model, the resulting discriminant functions are able to place data into the correct group with a 93.3% success rate.

QDA, with cross validation

```
alz.cv.qda <- qda(data[,c("hirecall","lirecall","hiunrem","liunrem","store","recog")], grouping=
data$group, CV=TRUE)
table(data$group, alz.cv.qda$class)
```

```
##
##      1  2  3
##   1 13  2  0
##   2  2  8  4
##   3  0  6 10
```

The MASS package also performs leave-one-out cross-validation. Less data is available to construct our model, and thus our classification results for quadratic discriminant analysis are less successful. Group 1 predictions were largely successful. Our model correctly predicted 13/15 group 1 individuals, with only 2/15 group 1 individuals being incorrectly placed into group 2. Group 2 predictions were less successful. The discriminant functions placed only 8/14 group 2 individuals correctly, with 6 group 2 individuals getting placed incorrectly. Group 3 predictions were roughly as unsuccessful as group 2 predictions. Our discriminant functions incorrectly placed 6 group 3 individuals into group 2. Contextualizing these results, it appears that the model is able to discern, with a 86.7% 'accuracy' rate, individuals with Alzheimer's from both the control group and individuals with depression. However, the model's biggest failure was in discerning depressed individuals from those who are not, with 28.6% truly depressed individuals being categorized as part of the control group. Finally, the cross-validated model seems to be able to discern individuals in the control group with an 62.5% 'accuracy' rate, but again, most of the error seems to arise from differentiating between the control group and those with depression. Overall, the discriminant functions of the cross-validated model are able to place data into the correct group with a 68.9% success rate. Though this number is markedly lower, it is probably a more accurate representation of the predictive power of our model.

6. Provide some evidence as to which of your original variables are the 'best' discriminators amongst your groups

```
DISCRIMINANT
 /GROUPS=group(1 3)
 /VARIABLES=hirecall lirecall hiunrem liunrem store recog
 /ANALYSIS ALL
 /SAVE=SCORES
 /METHOD=WILKS
 /FIN=3.84
 /FOUT=2.71
 /PRIORS EQUAL
 /HISTORY
 /STATISTICS=TABLE
 /CLASSIFY=NONMISSING SEPARATE.
```

Standardized Canonical Discriminant Function Coefficients

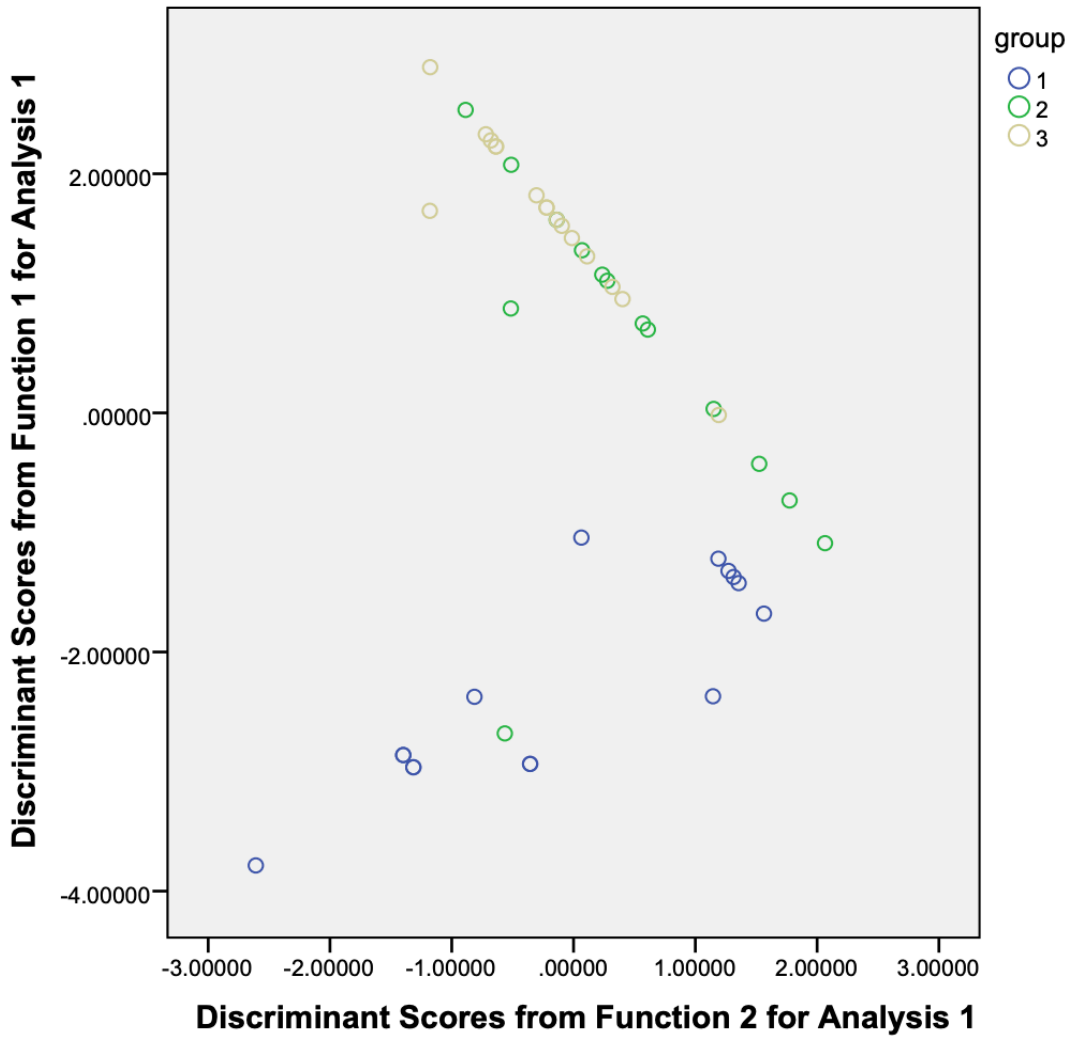
	Function	
	1	2
hiunrem	.818	-.666
recog	.372	.987

From our analysis, hiunrem (HI imagery unreminded memory) and recognition memory are two of the best discriminators out of our initial set of discriminator variables because those are the variables that are included in the functions following stepwise QDA. Above are the standardized discriminant function coefficients of each of the discriminator variables for each of the two discriminant functions. The magnitude of a standardized canonical discriminant function coefficient corresponds to the relative importance of discriminator variables . In this case the largest coefficients for DA function one is for hiunrem, while recog is a clear second place. As we saw earlier that the first DA function is significant and the second DA function is not, we conclude that hiunrem is a relatively better discriminator than recog.

If we had a significant second DA function, we would see that recog has the largest coefficient. Thus, if both DA functions were significant, the standardized coefficients of each variable would suggest that both were of relatively equal importance.

7. Make a score plot for the first two or three DA functions. Comment on what you see.

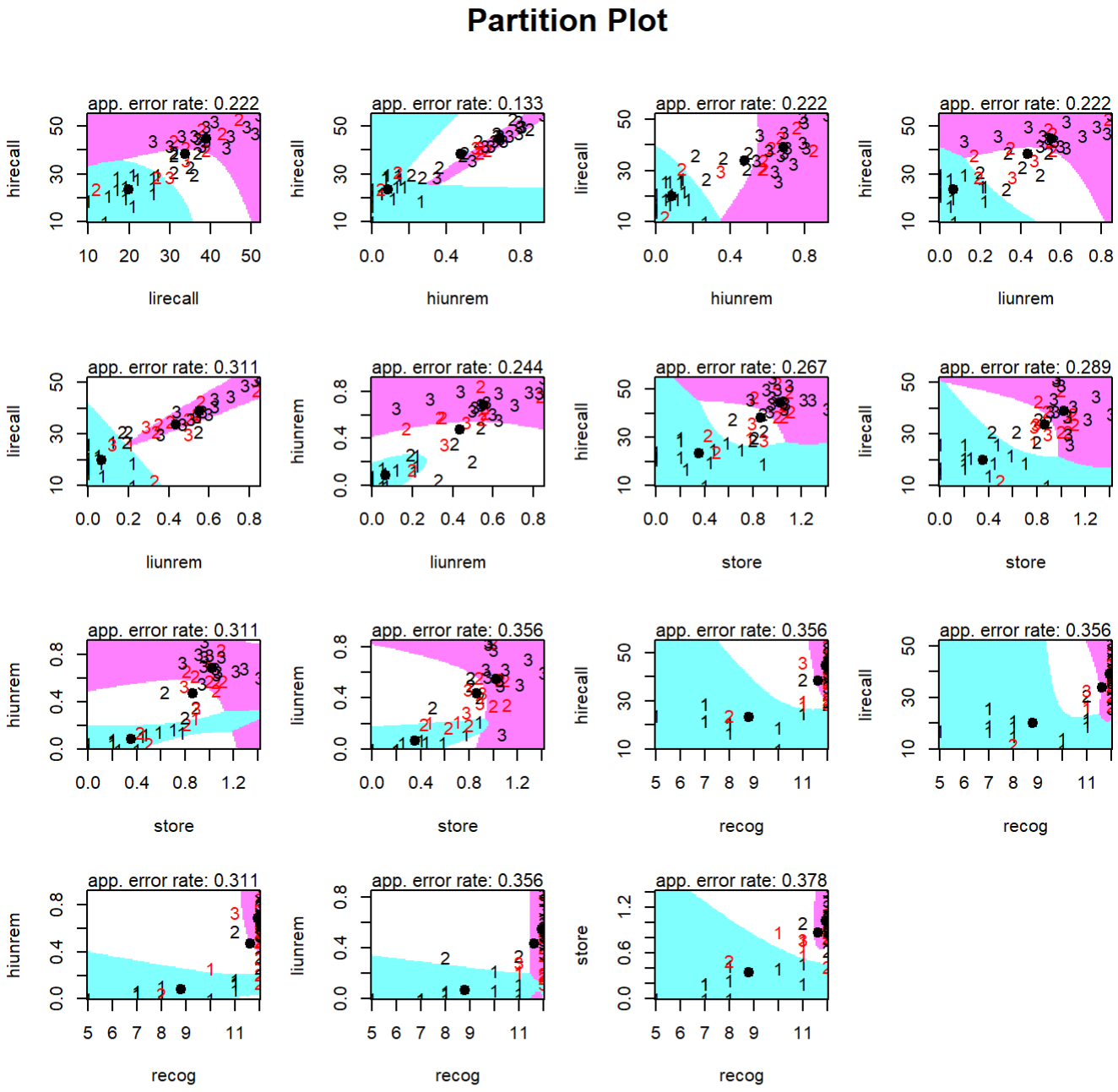
```
GRAPH
 /SCATTERPLOT(BIVAR)=Dis2_6 WITH Dis1_6 BY group
 /MISSING=LISTWISE.
```



Looking at the score plot for our DA function, we see that Group 1 is very clearly separated from group 2 and group 3. In addition, looking at the distribution, we begin to understand why it was challenging for our function to discriminate between 2 (depressed) and 3 (control). We note that there is a clustered line of group 2 and 3 together, where group one is relatively further away from 2 and 3, although we note that much of the group 3 points are on the upper portions of the line. This provides some insight as to why our function was strong at discriminating group 1, was relatively weak for group 3, and quite weak for group 2.

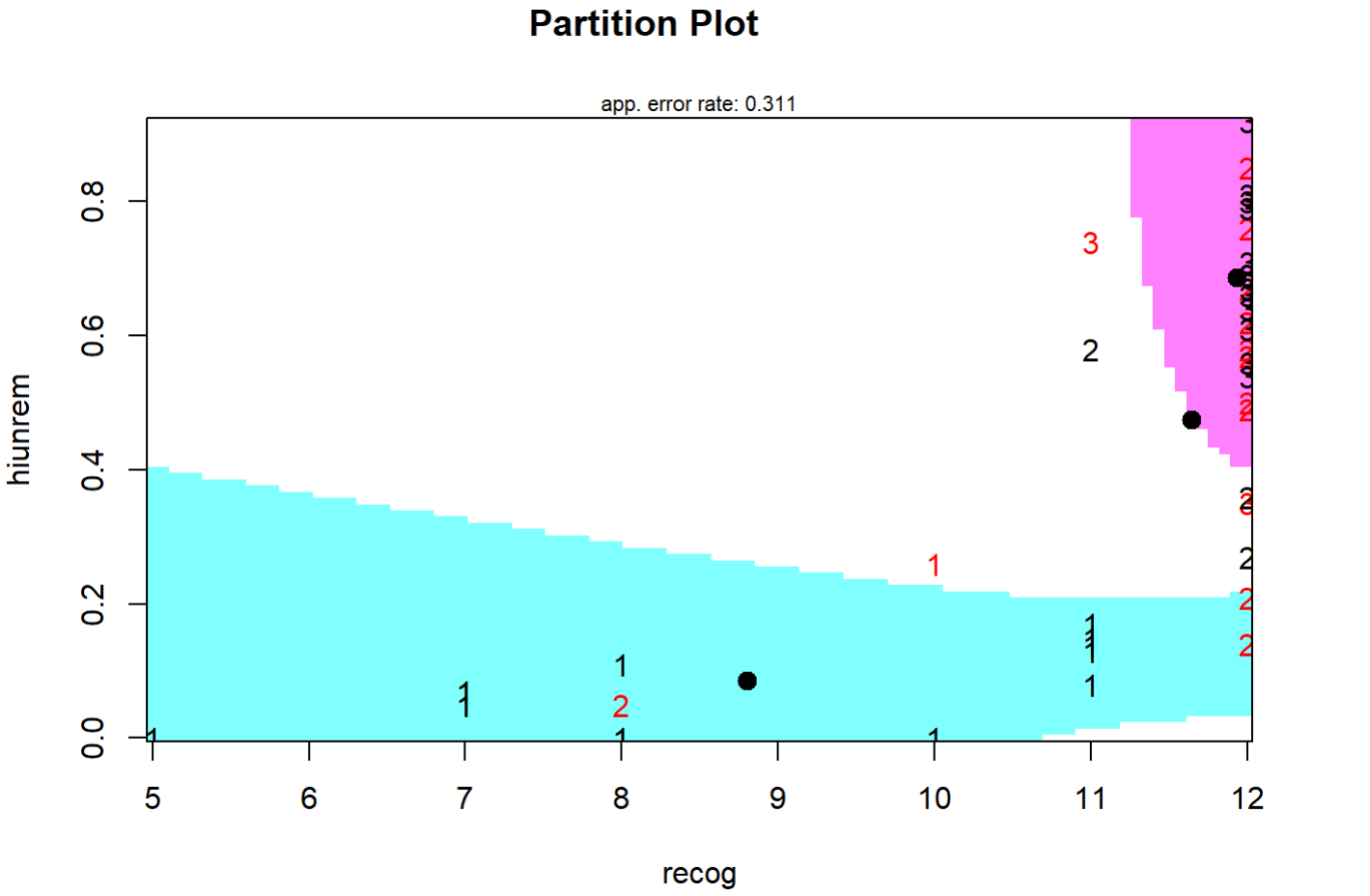
We also include Partition Plot below. We used R to get these plots.

```
partimat(as.factor(data$group)~hirecall+lirecall+hiunrem+liunrem+store+recog, data=data, method="qda")
```



And special attention to the partition of the variable space using the QDA variables, hiunrem and recog, that were identified as significant in Part 6

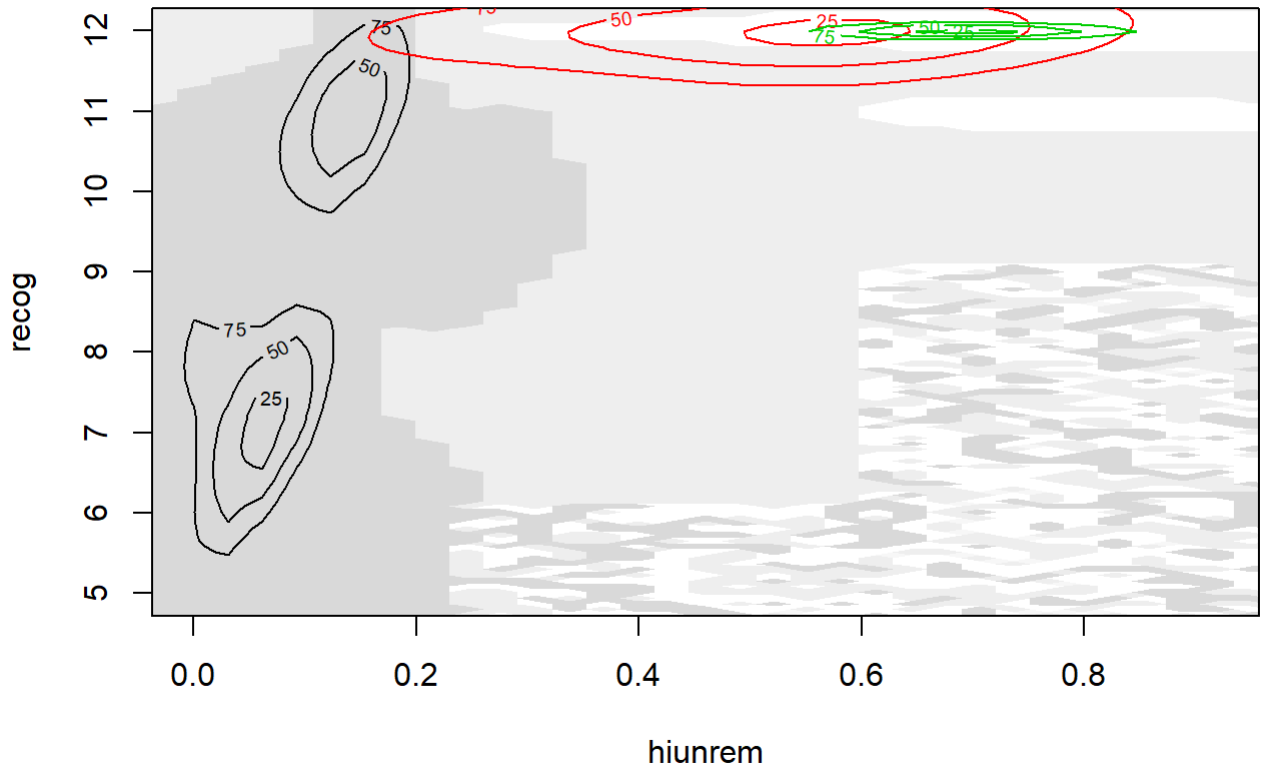
```
partimat(as.factor(data$group)~hiunrem+recog, data=data, method="qda")
```



8. Bonus (and optional)– try kernel smoothing or k-nearest neighbors and get the admiration of your professor and TA (and some extra credit)! You'll have to use SAS or R for this.

Part I: K-Density

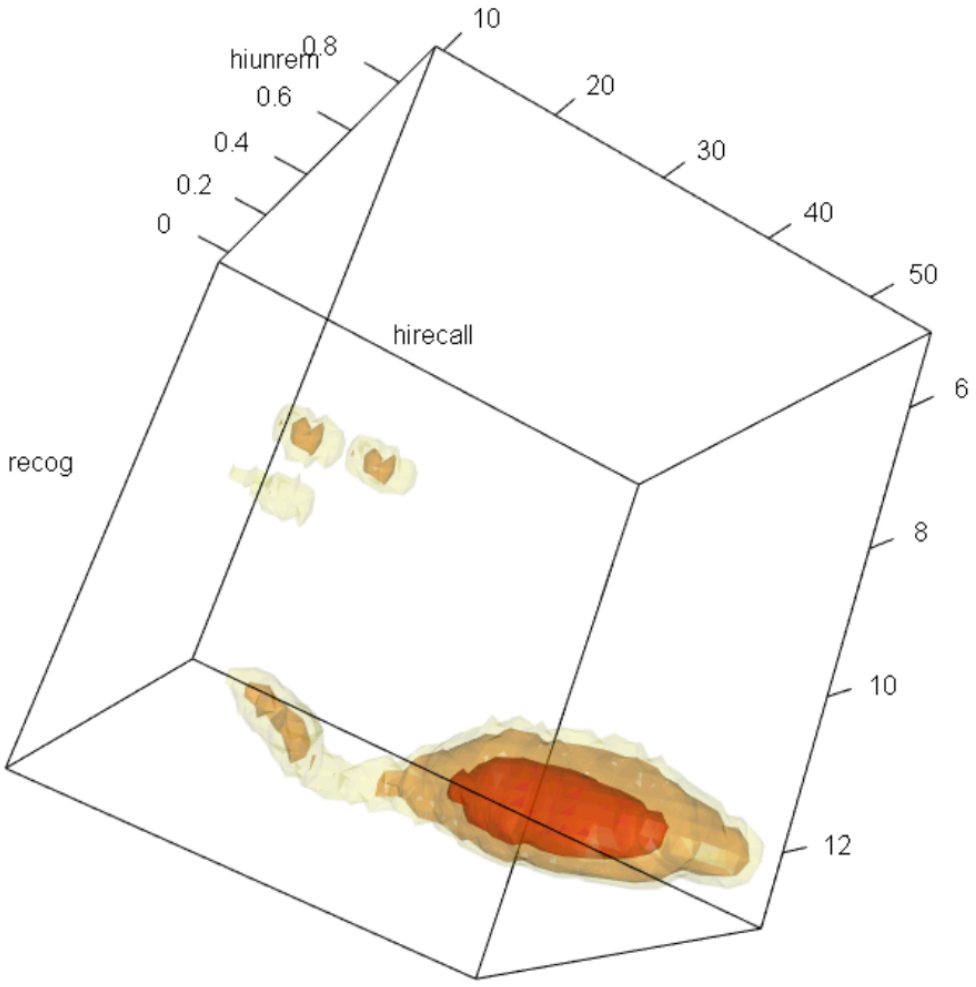
```
data.var <- data[,c(6,9)]
data.gr <- data[,10]
kda.fhat <- kda(x=data.var, x.group=data.gr)
plot(kda.fhat)
```



This is the kernel smoothing 2D map using the 'hiunrem' and 'recog' which identified as significant in Part 6. Black is group 1, red is group 2, green is group 3.

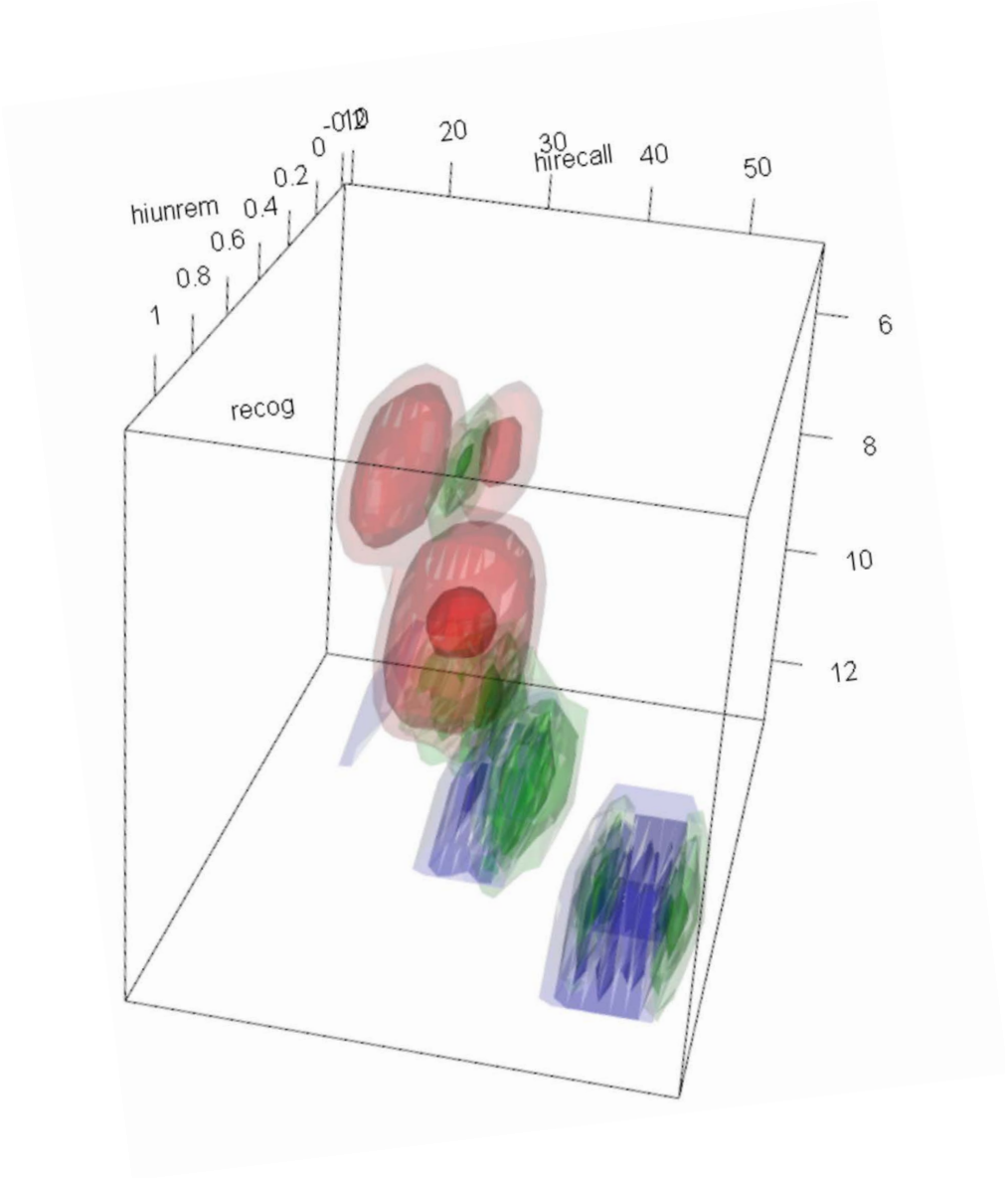
Recall that kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable. Kernel density estimation involves data smoothing problem where inferences about the population are made, based on a finite data set. Using R, we run kernel density estimation on recog, hirecall and hiunrem. To make space for different future interpretations, we wanted to create a 3D plot that visualises the distribution of data (probability density function) that's smoothed via KDE. We know that previously we identified record and hiunrem in our SPSS QDA analysis. In R, in stepwise, there are cases where hirecall is also considered significant, and it is the next best variable. Therefore, in 3-D, we used recog, hiunrem and hirecall. Below is the 3-D Kernel Smoothing.

```
H <- Hpi(x=alzheim[c(6,9,3)])
fhat <- kde(x=alzheim[c(6,9,3)], H=H)
plot(fhat, display="filled.contour2")
```

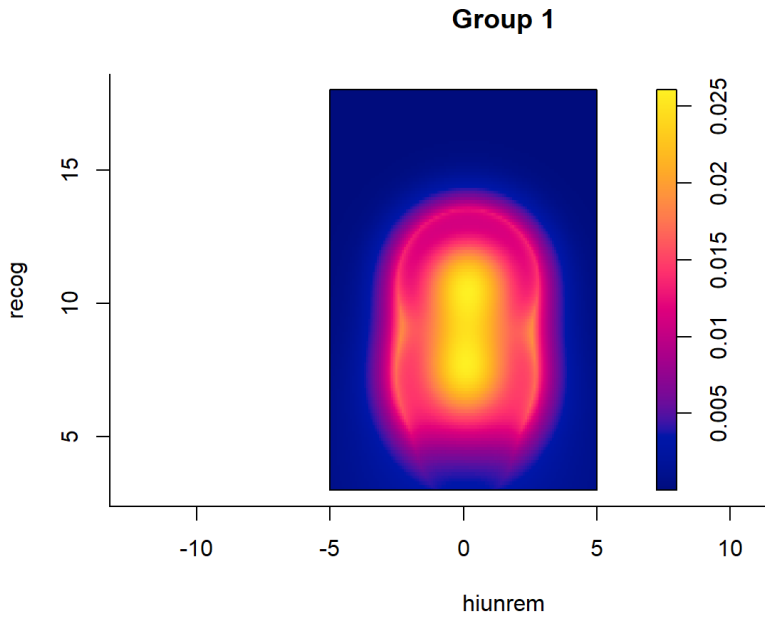


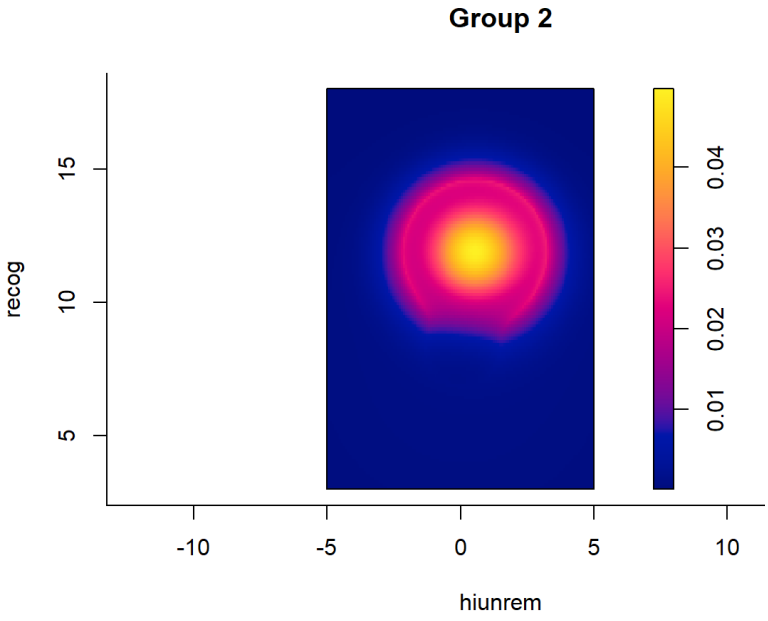
We also include a version coloured by groups. Red is group 1, green is group 2, and blue is group 3.

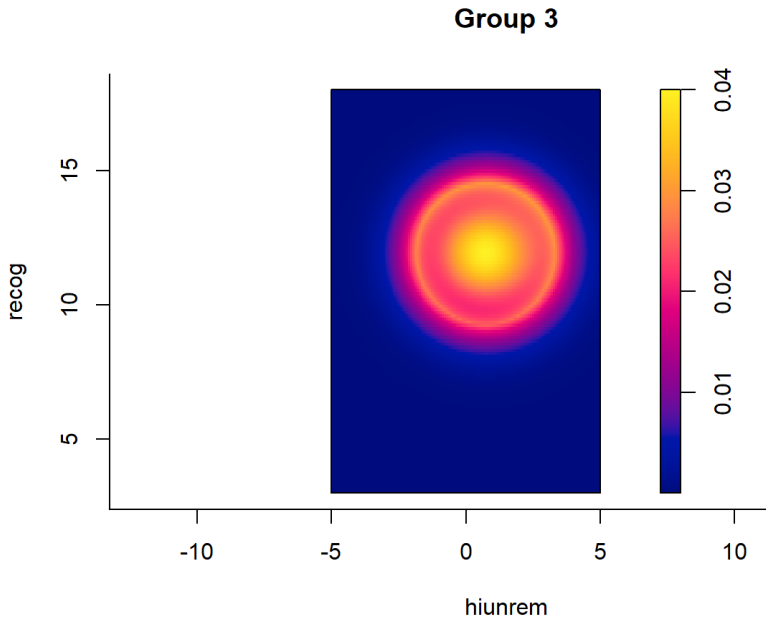
```
data.var <- data[,c(3,6,9)]
data.gr <- data[,10]
H <- Hkda(x=data.var, x.group=data.gr, bw="plugin")
kda.fhat <- kda(x=data.var, x.group=data.gr, Hs=H)
plot(kda.fhat)
```



```
x.1 <- jitter(data[data$group == 1,6],1)
y.1 <- jitter(data[data$group == 1,9],1)

x.2 <- jitter(data[data$group == 2,6],1)
y.2 <- jitter(data[data$group == 2,9],1)

x.3 <- jitter(data[data$group == 3,6],1)
y.3 <- jitter(data[data$group == 3,9],1)

pp.1 <- ppp(x.1, y.1, c(-5,5), c(3,18))
pp.2 <- ppp(x.2, y.2, c(-5,5), c(3,18))
pp.3 <- ppp(x.3, y.3, c(-5,5), c(3,18))

plot(bivariate.density(pp.1, h0=1.5, hp=1, adapt=TRUE), xlab="hiunrem", ylab="recog", main="Group 1")
plot(bivariate.density(pp.2, h0=1.5, hp=1, adapt=TRUE), xlab="hiunrem", ylab="recog", main="Group 2")
plot(bivariate.density(pp.3, h0=1.5, hp=1, adapt=TRUE), xlab="hiunrem", ylab="recog", main="Group 3")
```







This is adaptive kernel smoothing for each group, using Abramson's variable-bandwidth estimator as implemented in the 'sparr' package.

Part II: K-nearest neighbours

```
results <- matrix(nrow = 45, ncol = 20)
for (j in 1:20) {
  for (i in 1:45) {
    test_point <- data[i,]
    train_data <- data[-i,]
    knn_prediction <- as.vector(knn(train = train_data[,c(6,9)], test = test_point[,c(6,9)], cl
 = train_data[,10], k = j))
    truth <- as.vector(data[i,10])
    results[i,j] <- truth == knn_prediction
  }
}

best_k <- which.max(colMeans(results))
best_k
```

```
## [1] 9
```

```
success_rate <- colMeans(results)[best_k]
success_rate
```

```
## [1] 0.7555556
```

Leaving out one point, and predicting its group using the k-nearest neighbor method, and repeating this for all 45 points in our dataset, this method predicts the group of the unknown point with an approximately 75.6% success rate, using k=9 which was found to be the k with the best predictive power for our data.