PCA glmm

```
setwd("C:/Users/Helen/Desktop/Stats/Pruned3_big")

pcaData = read.csv("3factorPCA.csv")
```

Let's try removing outliers

```
PC1mean = mean(pcaData$PC1)
PC1sd = sd(pcaData$PC1)
PC2mean = mean(pcaData$PC2)
PC2sd = sd(pcaData$PC2)
PC3mean = mean(pcaData$PC3)
PC3sd = sd(pcaData$PC3)

pcaData = filter(pcaData, PC1 >= PC1mean - (2.5 * PC1sd))
pcaData = filter(pcaData, PC1 <= PC1mean + (2.5 * PC1sd))
pcaData = filter(pcaData, PC2 >= PC2mean - (2.5 * PC2sd))
pcaData = filter(pcaData, PC2 <= PC2mean + (2.5 * PC2sd))
pcaData = filter(pcaData, PC3 >= PC3mean - (2.5 * PC3sd))
pcaData = filter(pcaData, PC3 <= PC3mean + (2.5 * PC3sd))
```

```
pcaData$speaker = as.factor(pcaData$speaker)

m3 = glmer(label ~ PC1 + PC2 + PC3 + (1|speaker), data = pcaData, family=binomial)

r.squaredGLMM(m3)
```

```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.
```

```
## Warning: The null model is correct only if all variables used by the original
## model remain unchanged.
```

```
##                     R2m       R2c
## theoretical 0.10789239 0.3977706
## delta       0.09226173 0.3401445
```

```
summary(m3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: label ~ PC1 + PC2 + PC3 + (1 | speaker)
##    Data: pcaData
##
```

```
##      AIC      BIC   logLik deviance df.resid
##   6987.4   7020.5  -3488.7   6977.4     5541
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5309 -0.8583 -0.3503  0.8693  3.9519
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  speaker (Intercept) 1.584    1.258
## Number of obs: 5546, groups:  speaker, 12
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.85757    0.37104  -2.311   0.0208 *
## PC1          0.35593    0.01874  18.998  < 2e-16 ***
## PC2          0.01554    0.01857   0.837   0.4027
## PC3         -0.16700    0.02392  -6.980 2.95e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr) PC1    PC2
## PC1 -0.038
## PC2 -0.038  0.107
## PC3  0.037  0.025 -0.029
```

```
coefs = coef(m3)
coefs
```

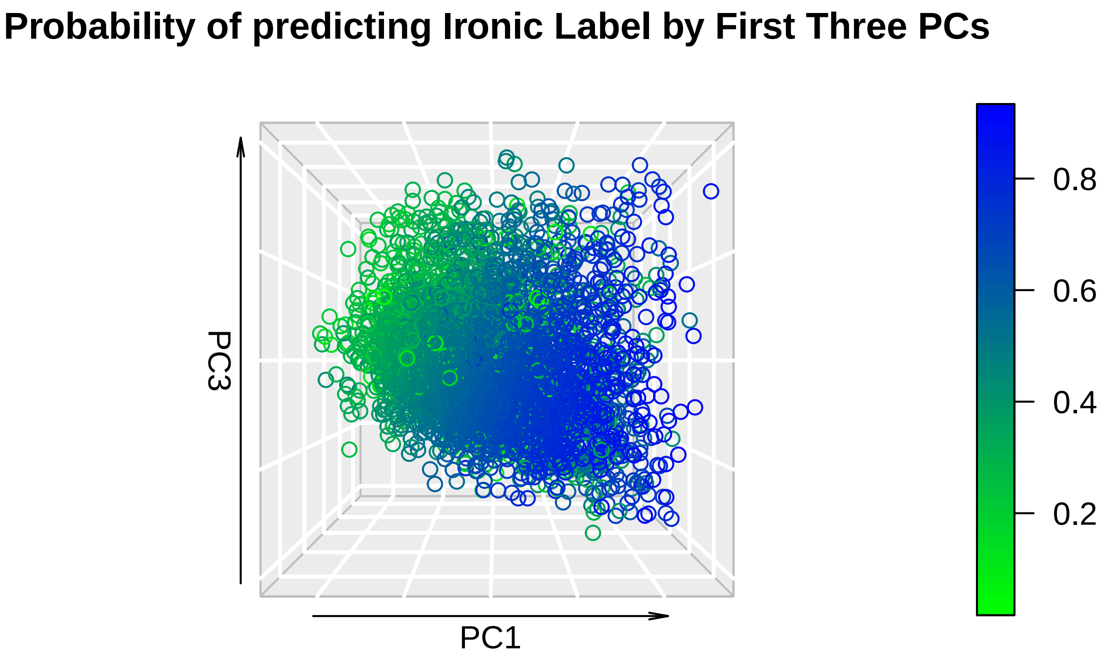
```
## $speaker
##   (Intercept)       PC1        PC2        PC3
## c  0.05517477 0.3559334 0.01553552 -0.1670019
## d  0.55343542 0.3559334 0.01553552 -0.1670019
## e -1.96344250 0.3559334 0.01553552 -0.1670019
## f -2.47981958 0.3559334 0.01553552 -0.1670019
## h -1.13184884 0.3559334 0.01553552 -0.1670019
## j -1.04611501 0.3559334 0.01553552 -0.1670019
## k  1.02870220 0.3559334 0.01553552 -0.1670019
## o -1.39473840 0.3559334 0.01553552 -0.1670019
## q -1.58285807 0.3559334 0.01553552 -0.1670019
## s -1.27280420 0.3559334 0.01553552 -0.1670019
## t  1.32014281 0.3559334 0.01553552 -0.1670019
## u -2.25778799 0.3559334 0.01553552 -0.1670019
##
## attr(,"class")
## [1] "coef.mer"
```

```
exp(coefs$speaker)
```

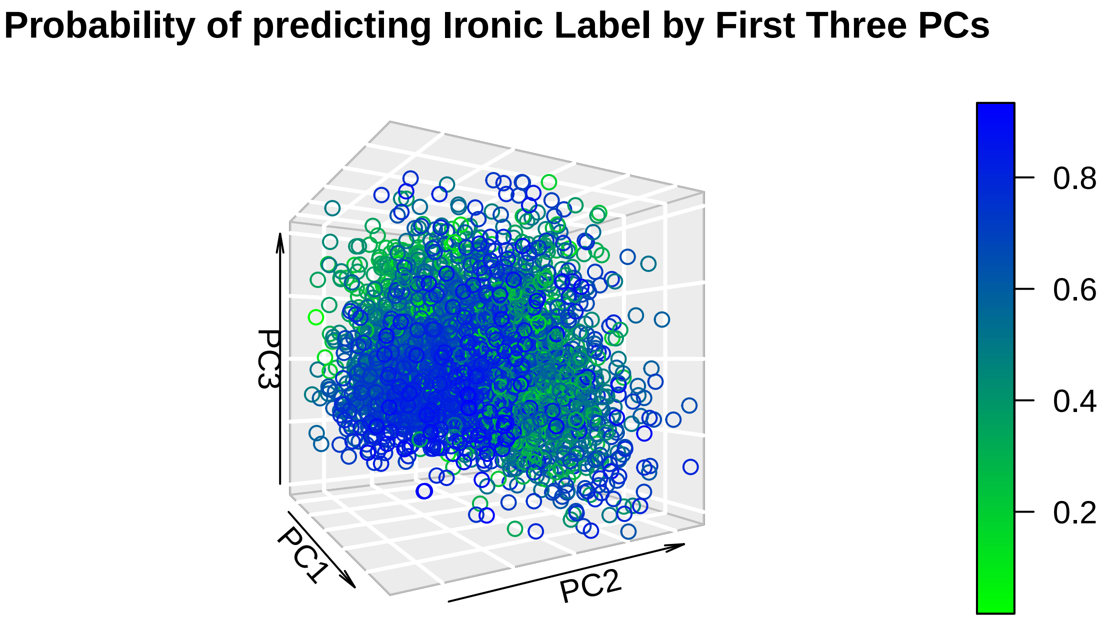
```
##   (Intercept)      PC1      PC2      PC3
## c  1.05672528 1.427512 1.015657 0.846198
## d  1.73921772 1.427512 1.015657 0.846198
## e  0.14037435 1.427512 1.015657 0.846198
## f  0.08375834 1.427512 1.015657 0.846198
## h  0.32243657 1.427512 1.015657 0.846198
## j  0.35129990 1.427512 1.015657 0.846198
## k  2.79743296 1.427512 1.015657 0.846198
## o  0.24789788 1.427512 1.015657 0.846198
## q  0.20538725 1.427512 1.015657 0.846198
## s  0.28004522 1.427512 1.015657 0.846198
## t  3.74395601 1.427512 1.015657 0.846198
## u  0.10458156 1.427512 1.015657 0.846198
```

```
pcaData$m3Fit = predict(m3, type="response")
iData = filter(pcaData, label==1)
nData = filter(pcaData, label==0)
```

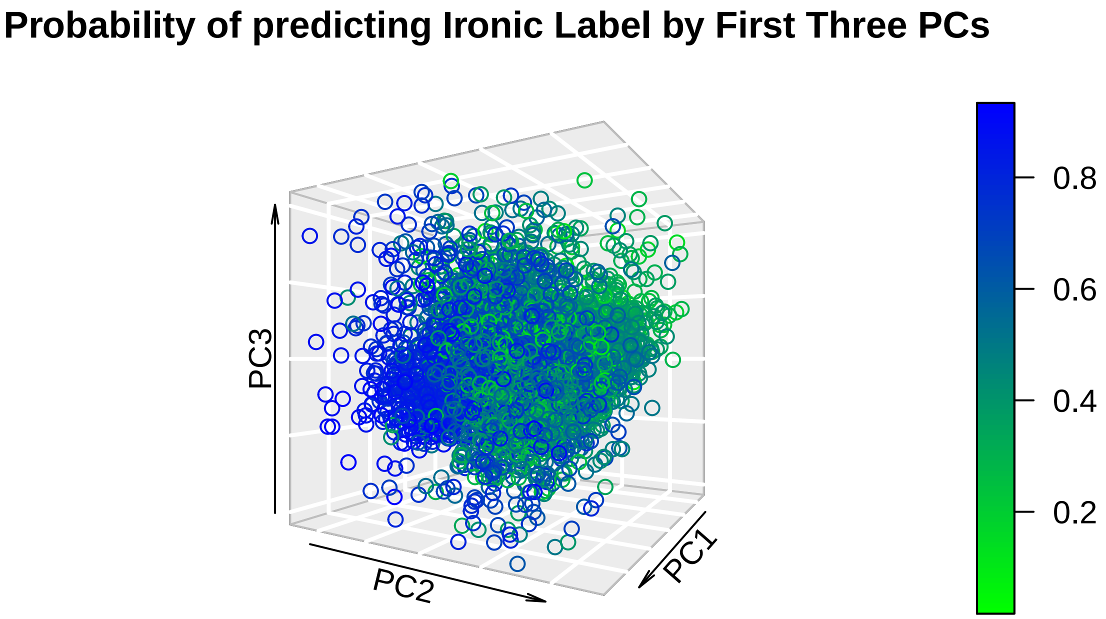
```
m3Plot1 = scatter3D(pcaData$PC1, pcaData$PC2, pcaData$PC3, phi = 0, theta = 0, bty="g",
                    colvar=pcaData$m3Fit, col=ramp.col(c("green", "blue")),
                    main = "Probability of predicting Ironic Label by First Three PCs",
                    xlab = "PC1", ylab = "PC2", zlab = "PC3")
```



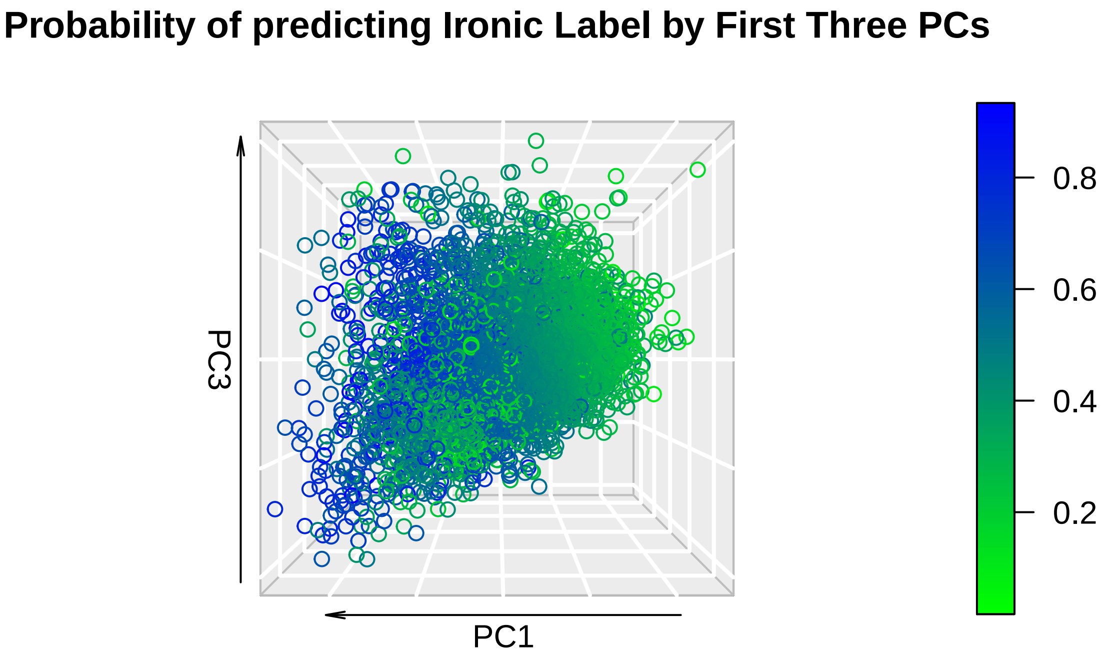
```
m3plot2 = scatter3D(pcaData$PC1, pcaData$PC2, pcaData$PC3, phi = 0, theta = 60,  bty="g",
                      colvar=pcaData$m3Fit, col=ramp.col(c("green", "blue")),
                      main = "Probability of predicting Ironic Label by First Three PCs",
                      xlab = "PC1", ylab = "PC2", zlab = "PC3")
```



```
m3Plot3 = scatter3D(pcaData$PC1, pcaData$PC2, pcaData$PC3, phi = 0, theta = 120,  bty="g",
                    colvar=pcaData$m3Fit, col=ramp.col(c("green", "blue")),
                    main = "Probability of predicting Ironic Label by First Three PCs",
                    xlab = "PC1", ylab = "PC2", zlab = "PC3")
```

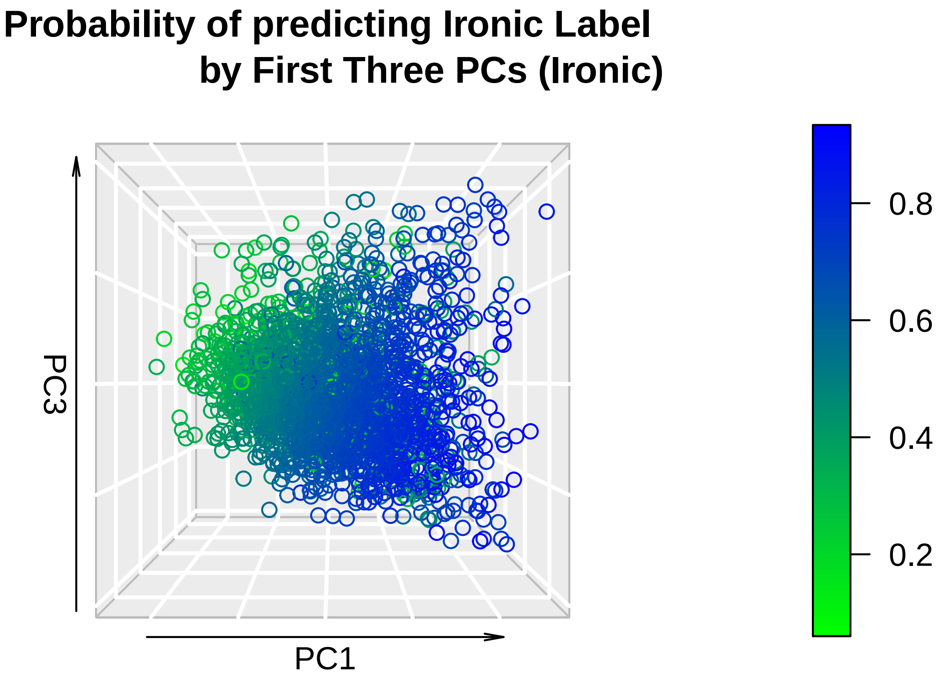


```
m3Plot4 = scatter3D(pcaData$PC1, pcaData$PC2, pcaData$PC3, phi = 0, theta = 180,  bty="g",
                    colvar=pcaData$m3Fit, col=ramp.col(c("green", "blue")),
                    main = "Probability of predicting Ironic Label by First Three PCs",
                    xlab = "PC1", ylab = "PC2", zlab = "PC3")
```

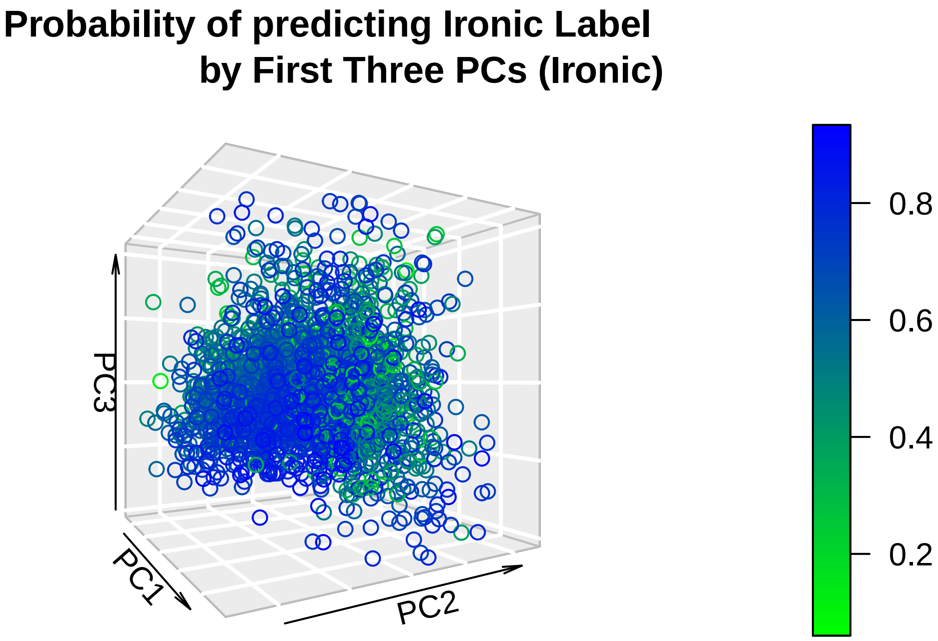


#Plotting only ironic samples

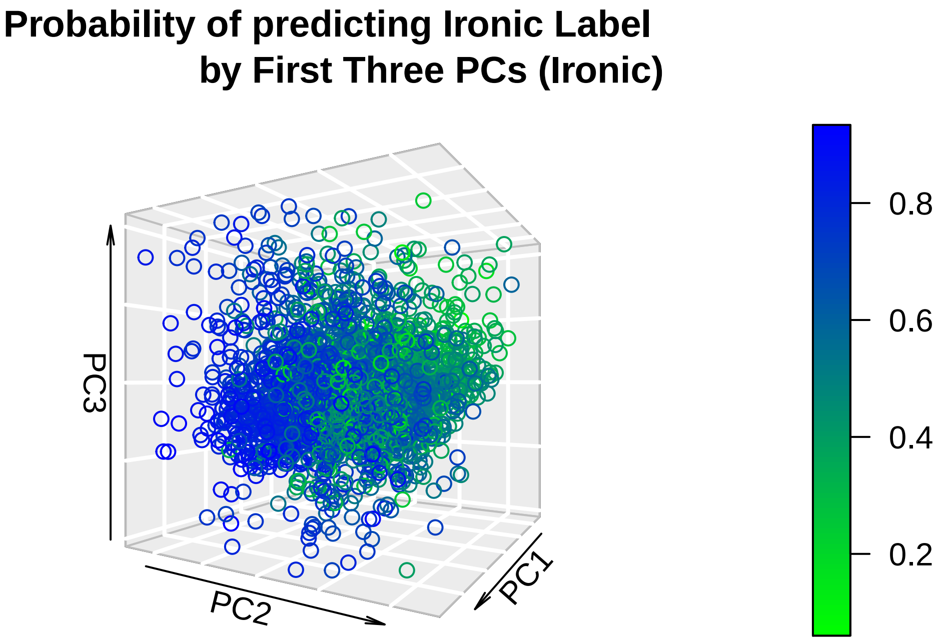
```
iPlot1 = scatter3D(iData$PC1, iData$PC2, iData$PC3, phi = 0, theta = 0,  bty="g",
                   colvar=iData$m3Fit, col=ramp.col(c("green", "blue")),
                   main = "Probability of predicting Ironic Label
                   by First Three PCs (Ironic)",
                   xlab = "PC1", ylab = "PC2", zlab = "PC3")
```



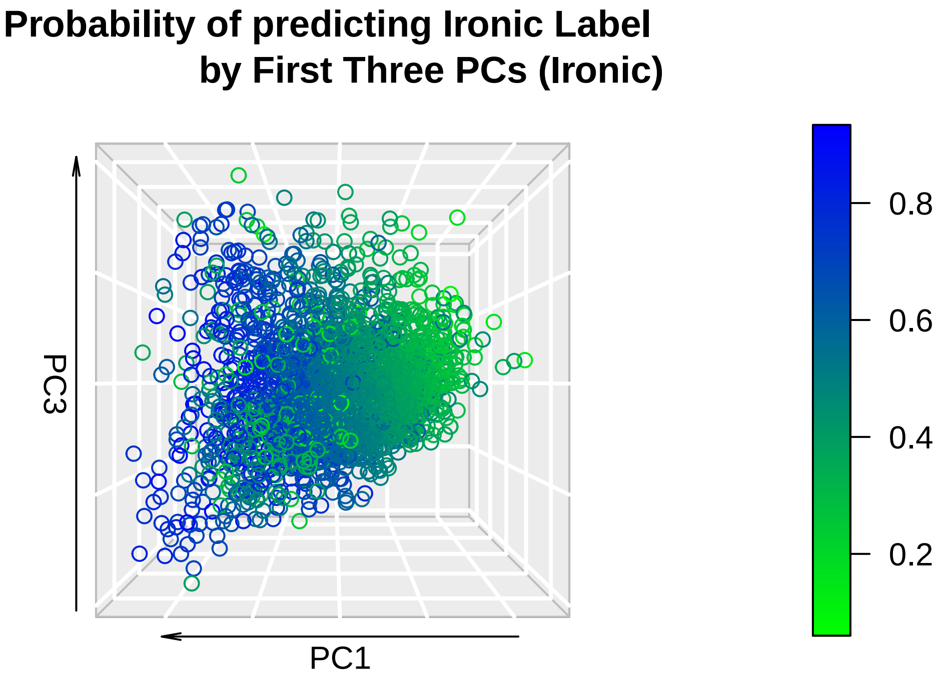
```
iPlot2 = scatter3D(iData$PC1, iData$PC2, iData$PC3, phi = 0, theta = 60,  bty="g",
                    colvar=iData$m3Fit, col=ramp.col(c("green", "blue")),
                    main = "Probability of predicting Ironic Label
                   by First Three PCs (Ironic)",
                    xlab = "PC1", ylab = "PC2", zlab = "PC3")
```



```
iPlot3 = scatter3D(iData$PC1, iData$PC2, iData$PC3, phi = 0, theta = 120,  bty="g",
                    colvar=iData$m3Fit, col=ramp.col(c("green", "blue")),
                    main = "Probability of predicting Ironic Label
                   by First Three PCs (Ironic)",
                    xlab = "PC1", ylab = "PC2", zlab = "PC3")
```

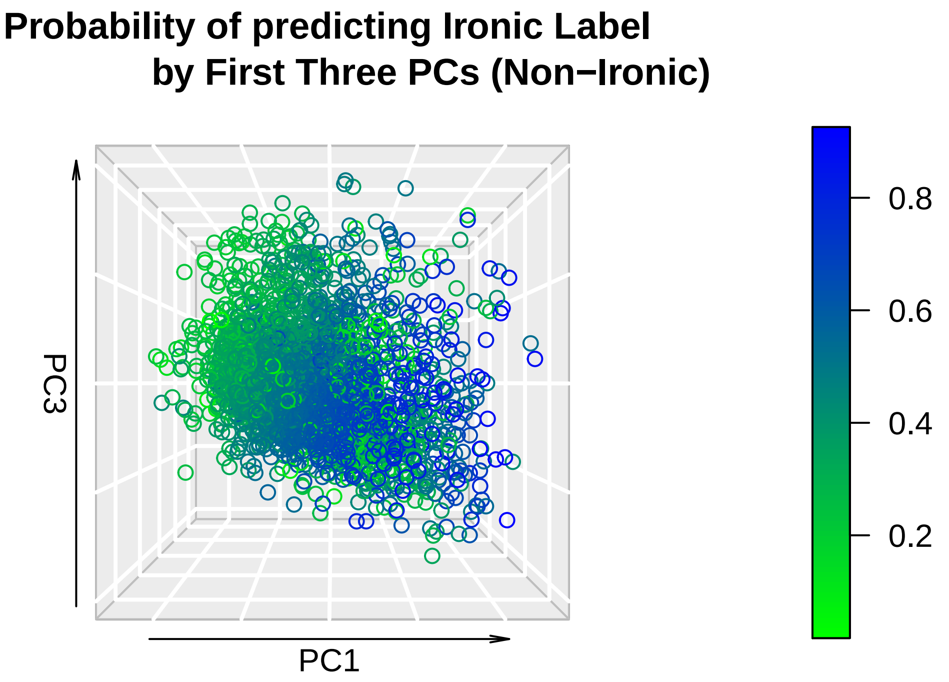


```
iPlot4 = scatter3D(iData$PC1, iData$PC2, iData$PC3, phi = 0, theta = 180,  bty="g",
                    colvar=iData$m3Fit, col=ramp.col(c("green", "blue")),
                    main = "Probability of predicting Ironic Label
                   by First Three PCs (Ironic)",
                    xlab = "PC1", ylab = "PC2", zlab = "PC3")
```

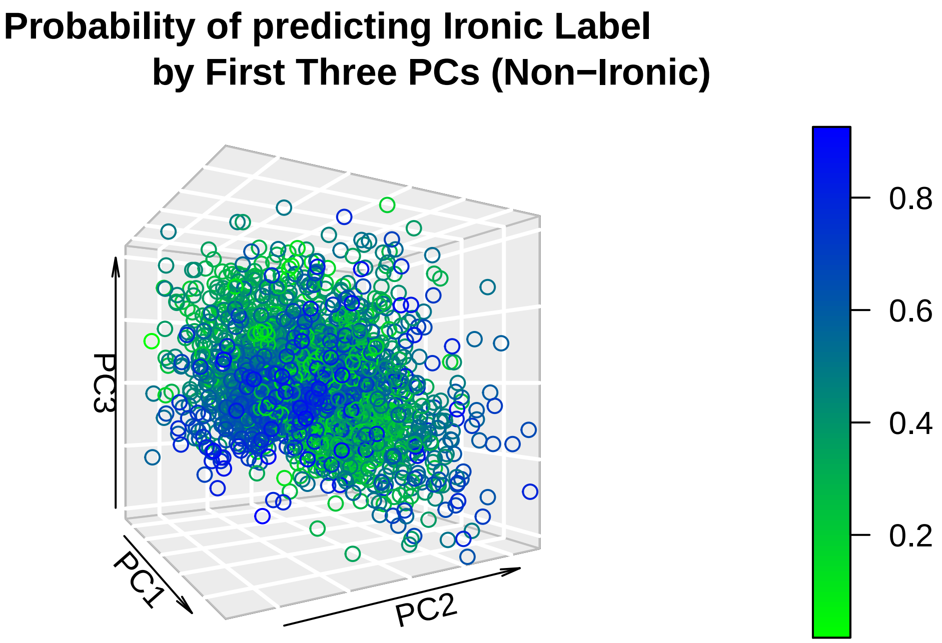


Plotting only non-ironic samples

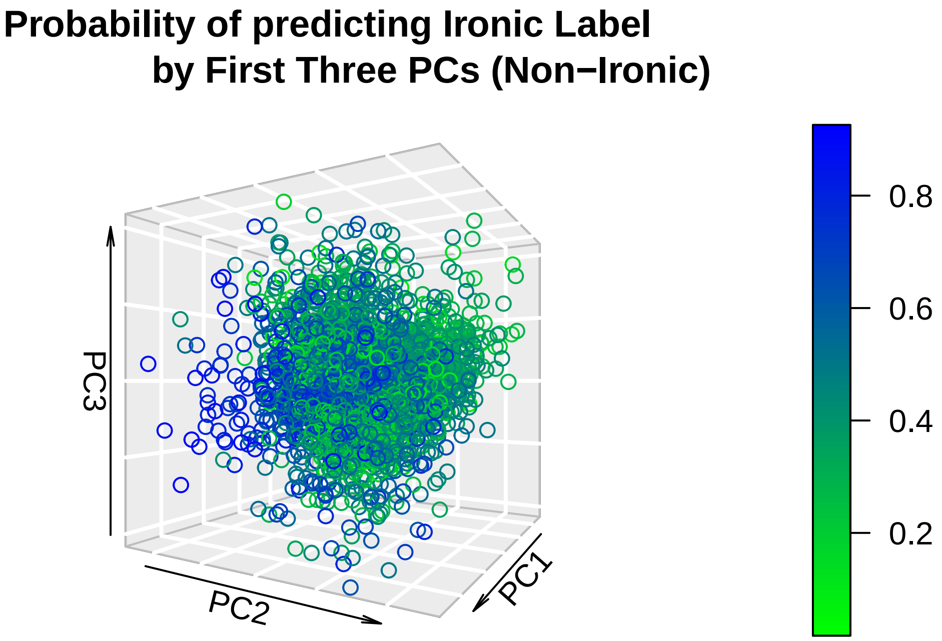
```
nPlot1 = scatter3D(nData$PC1, nData$PC2, nData$PC3, phi = 0, theta = 0,  bty="g",
                   colvar=nData$m3Fit, col=ramp.col(c("green", "blue")),
                   main = "Probability of predicting Ironic Label
                   by First Three PCs (Non-Ironic)",
                   xlab = "PC1", ylab = "PC2", zlab = "PC3")
```



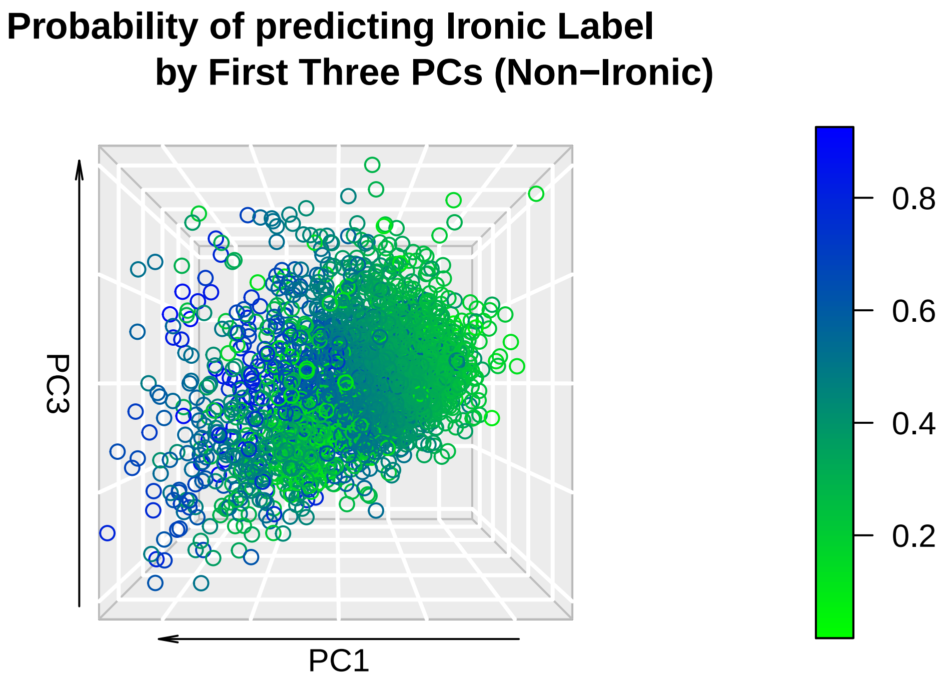
```
nPlot2 = scatter3D(nData$PC1, nData$PC2, nData$PC3, phi = 0, theta = 60,  bty="g",
                    colvar=nData$m3Fit, col=ramp.col(c("green", "blue")),
                    main = "Probability of predicting Ironic Label
                   by First Three PCs (Non-Ironic)",
                    xlab = "PC1", ylab = "PC2", zlab = "PC3")
```



```
nPlot3 = scatter3D(nData$PC1, nData$PC2, nData$PC3, phi = 0, theta = 120,  bty="g",
                    colvar=nData$m3Fit, col=ramp.col(c("green", "blue")),
                    main = "Probability of predicting Ironic Label
                   by First Three PCs (Non-Ironic)",
                    xlab = "PC1", ylab = "PC2", zlab = "PC3")
```



```
nPlot4 = scatter3D(nData$PC1, nData$PC2, nData$PC3, phi = 0, theta = 180,  bty="g",
                    colvar=nData$m3Fit, col=ramp.col(c("green", "blue")),
                    main = "Probability of predicting Ironic Label
                   by First Three PCs (Non-Ironic)",
                    xlab = "PC1", ylab = "PC2", zlab = "PC3")
```



Probability of predicting Ironic Label
by First Three PCs (Non−Ironic)

Fresh model trained on 80% of the data and tested on 20% to get ROC curve

Remove model predictions from original data

```
pcaData = select(pcaData, -m3Fit)
```

Train/test split

```
smp_size = floor(0.80 * nrow(pcaData))

set.seed(6)
train_ind = sample(seq_len(nrow(pcaData)), size = smp_size)

trainData = pcaData[train_ind, ]
testData = pcaData[-train_ind, ]
```

Train fresh model on training data

```
#all effects without interaction term
m4 = glmer(label ~ PC1 + PC2 + PC3 + (1|speaker),
           data = trainData, family=binomial)
```

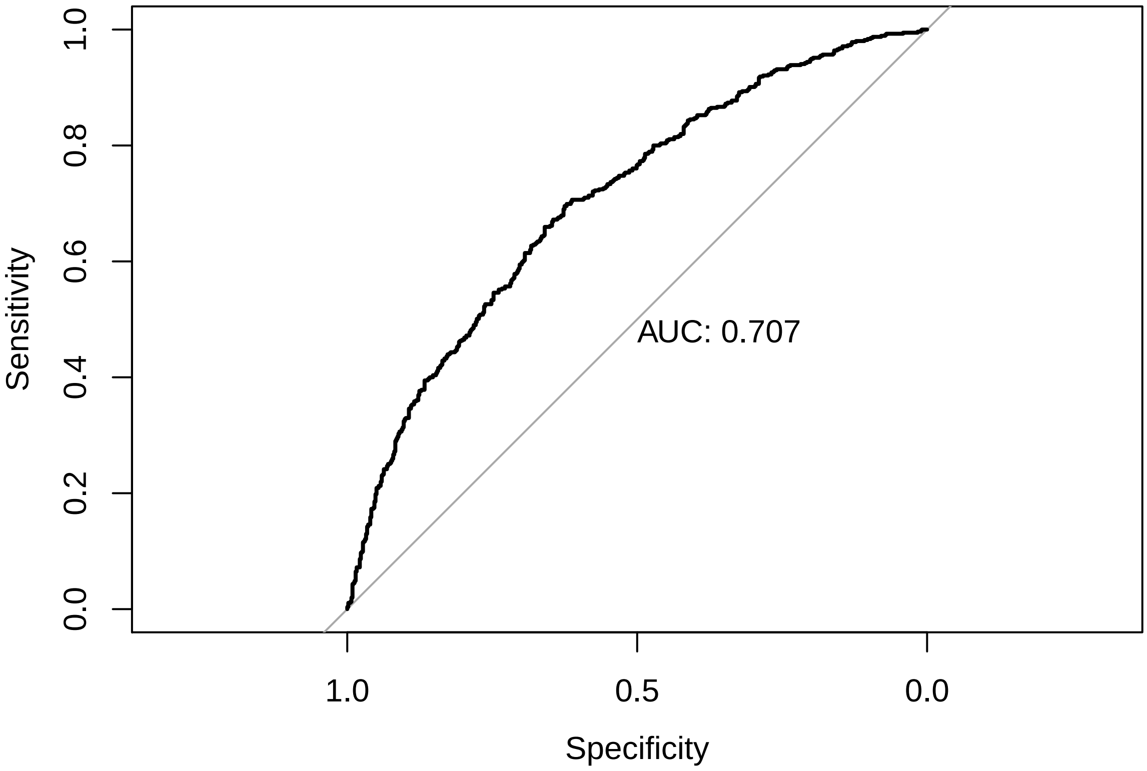
Calculate predictions from test data

```
testData$m4pred = predict(m4, testData, type="response")
```

```
test_roc = roc(testData$label ~ testData$m4pred, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



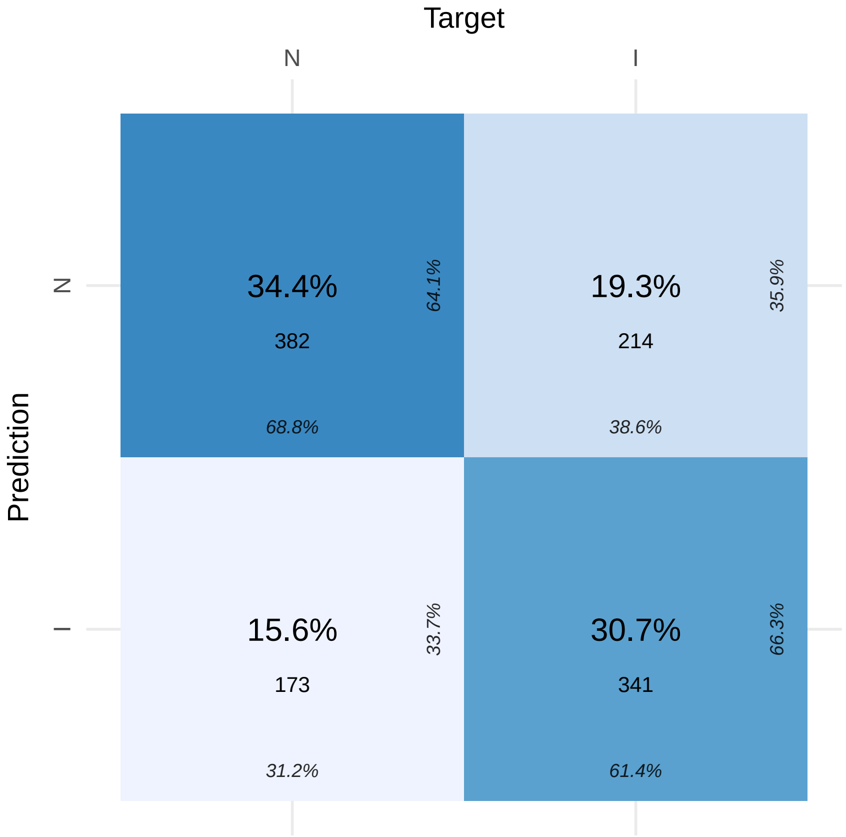
```
eval = evaluate(testData, target_col = "label",
                prediction_cols = "m4pred", type = "binomial")
cm = eval$‘Confusion Matrix‘[[1]]

cm$Prediction = c("N", "I", "N", "I")
cm$Target = c("N", "N", "I", "I")

eval
```

```
## # A tibble: 1 x 19
##   ‘Balanced Accur~ Accuracy    F1 Sensitivity Specificity ‘Pos Pred Value‘
##              <dbl>    <dbl> <dbl>       <dbl>       <dbl>            <dbl>
## 1            0.651    0.651 0.638       0.614       0.688            0.663
## # ... with 13 more variables: ‘Neg Pred Value‘ <dbl>, AUC <dbl>, ‘Lower
## #   CI‘ <dbl>, ‘Upper CI‘ <dbl>, Kappa <dbl>, MCC <dbl>, ‘Detection
## #   Rate‘ <dbl>, ‘Detection Prevalence‘ <dbl>, Prevalence <dbl>,
## #   Predictions <list>, ROC <named list>, ‘Confusion Matrix‘ <list>,
## #   Process <list>
```

```
plot_confusion_matrix(cm)
```

```
## Warning in plot_confusion_matrix(cm): ’ggimage’ is missing. Will not plot arrows
## and zero-shading.
```

```
## Warning in plot_confusion_matrix(cm): ’rsvg’ is missing. Will not plot arrows
## and zero-shading.
```


Target
N
I
Prediction
N
I