

Models

```
pacman::p_load(R.matlab, dplyr, caret, factoextra, FactoMineR
)
```

Data Setup/ Import

```
# Import mouse 251 data
bb <- readMat("Data/Zero_Maze/608034_409/Day_1/Trial_001_0/binned_behavior.mat")

## Warning in readMat("Data/Zero_Maze/608034_409/Day_1/Trial_001_0/
## binned_behavior.mat"): strings not representable in native encoding will be
## translated to UTF-8

bb <- data.frame(t(bb$binned.behavior))
names(bb) <- c("open", "closed")

bz <- readMat("Data/Zero_Maze/608034_409/Day_1/Trial_001_0/binned_zscore.mat")
bz <- data.frame(bz$binned.zscore)

data <- data.frame(cbind(bb, bz))
data$time <- seq(1:nrow(data))

data_clean <- data[-which(data$open==0 & data$closed==0),]

rm(bb, bz, data)
```

Logistic Model with Random Neurons

```
mdl_data <- data_clean %>% select(-closed)

# a <- NULL
# for (i in 1:100){
#   for (s in c(1,5,25,50,100)){
#     sample <- sample(2:ncol(mdl_data), size = s)
#     train <- sample(c(TRUE, FALSE), size = nrow(mdl_data), replace = TRUE)
#     train_data <- mdl_data[train, c(1,sample)]
#     test_data <- mdl_data[!train, c(1,sample)]
#     mdl <- glm(open~., data = train_data, family = binomial(link = "logit"))
#     pred <- predict(mdl, test_data, type = "response")
#     pred <- factor(ifelse(pred>=0.5, 1, 0), levels = c(1,0))
#     cm <- confusionMatrix(data = pred, reference = factor(test_data$open, levels = c(1,0)))
```

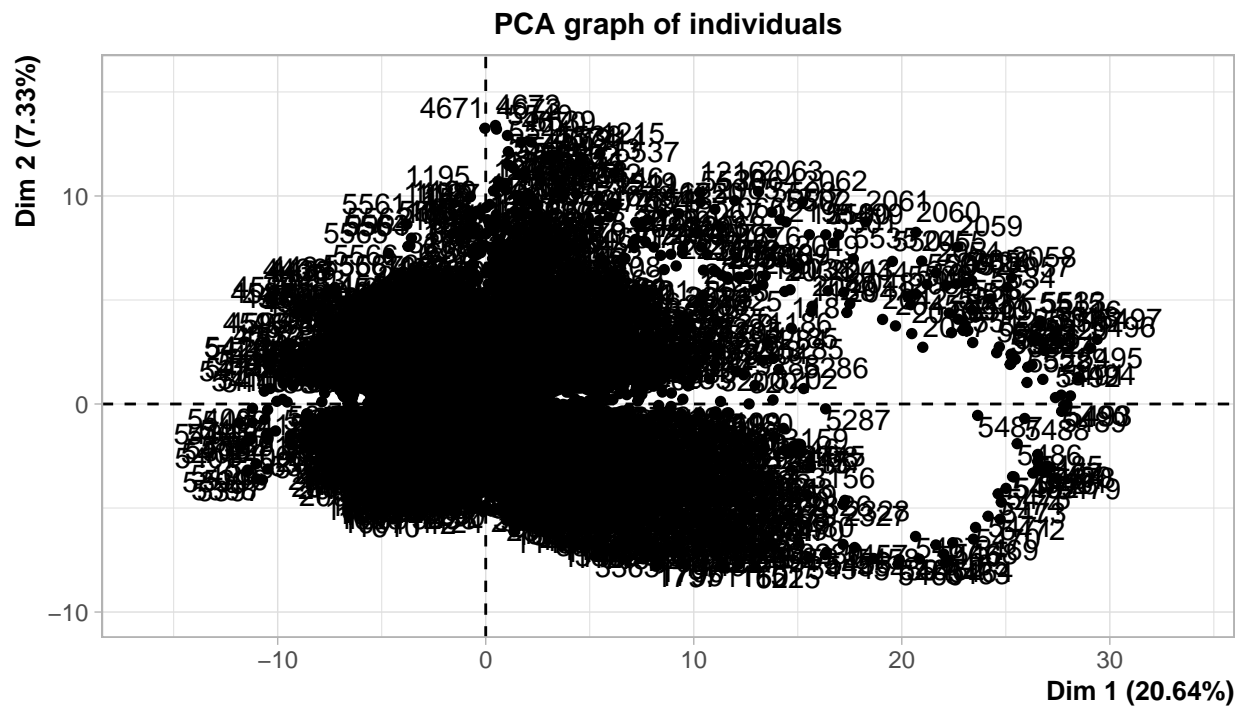
```

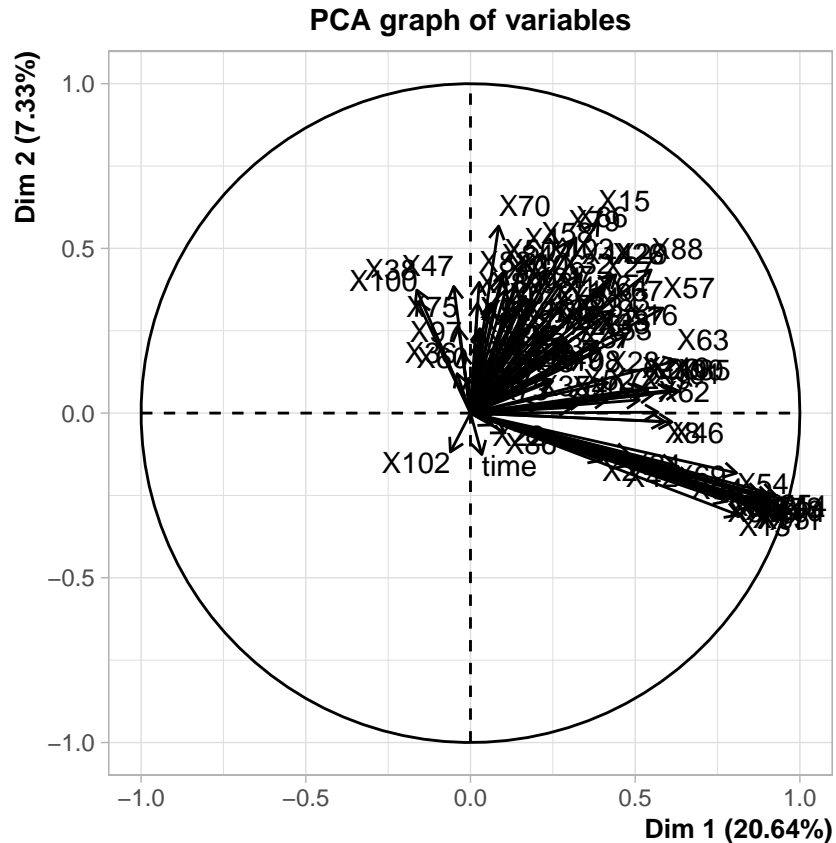
# a <- rbind(a,c(cm$overall["Accuracy"], s) )
# }
# #print(i)
# }
# a <- data.frame(a)
#
# ggplot(data = a)+
#   geom_histogram(mapping = aes(x = Accuracy))+
#   facet_wrap(vars(V2))
#
# means <- a %>% group_by(V2) %>% summarise(mean = mean(Accuracy))
# means[which.max(means$mean),]

```

Logistic Model Combined With PCA

```
mdl_data.pca <- PCA(mdl_data[,-1], scale.unit = TRUE, ncp=20, graph = TRUE)
```





```
eigenvalue <- get_eigenvalue(mdl_data.pca) #80:95.6%; 60:90%; 40: 80.8%
# extract principal components
comp <- data.frame(mdl_data.pca$ind$coord)
# fit model
comp$open <- mdl_data$open
train <- sample(1:nrow(comp), size = round(.8*nrow(comp)), replace = FALSE)
training <- comp[train,]
testing <- comp[-train,]
mdl_logis <- glm(open~., data = training, family = binomial("logit"))
summary(mdl_logis)
```

```
##
## Call:
## glm(formula = open ~ ., family = binomial("logit"), data = training)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.04634  -0.05288   0.32621   0.57871   2.46619
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.728854   0.053119  32.547  < 2e-16 ***
## Dim.1        -0.028109   0.009346  -3.008  0.00263 **
## Dim.2         0.055512   0.017007   3.264  0.00110 **
## Dim.3        -0.517903   0.023124 -22.396  < 2e-16 ***
```

```

## Dim.4      0.307133  0.021397  14.354 < 2e-16 ***
## Dim.5      0.369732  0.024980  14.801 < 2e-16 ***
## Dim.6      0.013305  0.028056   0.474 0.63533
## Dim.7      0.074841  0.029347   2.550 0.01077 *
## Dim.8     -0.418069  0.030125 -13.878 < 2e-16 ***
## Dim.9     -0.010492  0.029703  -0.353 0.72393
## Dim.10     0.204702  0.032020   6.393 1.63e-10 ***
## Dim.11     0.344645  0.032671  10.549 < 2e-16 ***
## Dim.12    -0.525379  0.037316 -14.079 < 2e-16 ***
## Dim.13    -0.083059  0.033354  -2.490 0.01277 *
## Dim.14     0.122502  0.037265   3.287 0.00101 **
## Dim.15     0.360993  0.039287   9.189 < 2e-16 ***
## Dim.16     0.172321  0.038033   4.531 5.87e-06 ***
## Dim.17    -0.017017  0.039121  -0.435 0.66357
## Dim.18    -0.019319  0.038545  -0.501 0.61622
## Dim.19     0.300332  0.039207   7.660 1.86e-14 ***
## Dim.20    -0.012899  0.039037  -0.330 0.74107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 5564.3  on 4946  degrees of freedom
## Residual deviance: 3764.3  on 4926  degrees of freedom
## AIC: 3806.3
##
## Number of Fisher Scoring iterations: 6

pred_logis <- predict(mdl_logis, newdata = testing)
error_rate <- mean((pred_logis>.5 & testing$open == 0) | (pred_logis<.5 & testing$open == 1))
error_rate

## [1] 0.1980598

```