"Full Data" Analysis Script (Mixed Models)

Paula McMahon

August 2019

```
knitr::opts_chunk$set(echo = TRUE)
library(readxl)
library(corrplot)
library(tree)
library(randomForest)
library(e1071)
library(MASS)
library(moments)
library(rpart)
library(rpart.plot)
library(class)
library(caret)
library(gbm)
library(lme4)
library(lubridate)
library(npmlreg)
```

Read in data

```
# Read in DORMANT customer data
dormant_customers <-
  read_excel("~/college/ST606_Project/data_files/dormant-transactions.xlsx")

# Read in CURRENT customer data
current_customers <-
  read_excel("~/college/ST606_Project/data_files/stayed-shopping-transactions.xlsx")
```

Merge current and dormant datasets.

```
# Add a variable "Churn" to the dormant dataset and set it equal to 1
# i.e. the customer has stopped shopping.
# Add a variable "Churn" to the current dataset and set it equal to 0
# i.e. the customer is a current shopper.
current_customers$Churn = 0
dormant_customers$Churn = 1

# Not all of the variables in the dataset will be used:
# OrderRef is a unique reference number and is not of any value.
# ExpectedGoodsCharge is 98% correlated with ActualCharge, so drop
# ExpectedGoodsCharge from analysis.
# OverallSubstitutionPolicy is a categorical variable and was not used in
# the aggregated analysis but can be used in the full analysis.
# TotalOrderLines is an extra variable in current dataset but not dormant,
# so drop from analysis.
```

```r
# TotalItemsApprovedPicks is 100% correlated with TotalPickedLines so
# drop TotalItemsApprovedPicks from analysis
# TotalQtyOrdered is 99% correlated with TotalOrderItems so drop
# TotalQtyOrdered from analysis.
# AvailabilityPostSubPercentage is 99.9% correlated with
# AvailabilityPreSubPercentage so drop AvailabilityPostSubPercentage from analysis.

merged_customers <-
  rbind(dormant_customers[c("sequenceID","StoreID","ExpectedFulfillmentCharge",
                            "OverallSubstitutionPolicy","SlotStartDate",
                            "ActualCharge","TotalOrderItems","TotalPickedLines",
                            "TotalQtySubbed","TotalQtyOOS","TotalPickTimeSeconds",
                            "AvailabilityPreSubPercentage","PercentageOutOfStocks",
                            "PercentageSubstitutions","ValueOfSubstitutions",
                            "ValueOfOutOfStocks","RelatedCallsCount","Churn")],
        current_customers[c("sequenceID","StoreID","ExpectedFulfillmentCharge",
                            "OverallSubstitutionPolicy","SlotStartDate",
                            "ActualCharge","TotalOrderItems","TotalPickedLines",
                            "TotalQtySubbed","TotalQtyOOS","TotalPickTimeSeconds",
                            "AvailabilityPreSubPercentage","PercentageOutOfStocks",
                            "PercentageSubstitutions","ValueOfSubstitutions",
                            "ValueOfOutOfStocks","RelatedCallsCount","Churn")])
```

```r
# Create factors
merged_customers$Churn <- as.factor(merged_customers$Churn)
merged_customers$StoreID <- as.factor(merged_customers$StoreID)
merged_customers$OverallSubstitutionPolicy <-
  as.factor(merged_customers$OverallSubstitutionPolicy)
dim(merged_customers)
```

Create time variables

```r
merged_customers$Month = numeric(30977)
merged_customers$Month <- month(as.POSIXlt(merged_customers$SlotStartDate,
                                           format="%d/%m/%Y"))

merged_customers$Season = character(30977)
merged_customers$Season <- month(as.POSIXlt(merged_customers$SlotStartDate,
                                            format="%d/%m/%Y"))
for (i in 1:nrow(merged_customers))
# 1 = February, March, April (SPRING)
ifelse(merged_customers$Season[i] == 2 | merged_customers$Season[i] == 3 |
         merged_customers$Season[i] == 4, merged_customers$Season[i] <- "Spring",
# 2 = May, June, July (SUMMER)
ifelse(merged_customers$Season[i] == 5 | merged_customers$Season[i] == 6 |
         merged_customers$Season[i] == 7, merged_customers$Season[i] <- "Summer",
# 3 = August, September, October (AUTUMN)
ifelse(merged_customers$Season[i] == 8 | merged_customers$Season[i] == 9 |
         merged_customers$Season[i] == 10, merged_customers$Season[i] <- "Autumn",
# 4 = November, December, January (WINTER)
merged_customers$Season[i] <- "Winter")))
```

```r
# Make Season and Month factors
merged_customers$Season <- as.factor(merged_customers$Season)
merged_customers$Month <- as.factor(merged_customers$Month)
# Change the reference level of the factor so that the 5th Month is the reference level
merged_customers$Month <- relevel(merged_customers$Month, ref = 5)
```

Data Scale and Split

```r
# Scale the data as predictor variables are on very different scales.
# These are the predictor variables to scale
pvars <- c("ExpectedFulfillmentCharge","ActualCharge","TotalOrderItems",
"TotalPickedLines","TotalQtySubbed","TotalQtyOOS","TotalPickTimeSeconds",
"AvailabilityPreSubPercentage","PercentageOutOfStocks","PercentageSubstitutions",
"ValueOfSubstitutions","ValueOfOutOfStocks","RelatedCallsCount")
merged_customers_sc <- merged_customers
merged_customers_sc[pvars] <- lapply(merged_customers_sc[pvars],scale)

set.seed(123)
s <- sample(nrow(merged_customers_sc), round(.5*(nrow(merged_customers_sc))))
merged_customers_sc_train <- merged_customers_sc[s,]     # training set
merged_customers_sc_test  <- merged_customers_sc[-s,]    # test set
dim(merged_customers_sc_train)
dim(merged_customers_sc_test)
```

Logistic Regression Mixed Models

Model 1

```r
# random effect : sequenceID
# fixed effects : all other predictors
fitlrmm1 <- glmer(Churn ~ ExpectedFulfillmentCharge+ActualCharge+TotalOrderItems+
                   TotalPickedLines+TotalQtySubbed+TotalQtyOOS+TotalPickTimeSeconds+
                   AvailabilityPreSubPercentage+PercentageOutOfStocks+
                   PercentageSubstitutions+ValueOfSubstitutions+ValueOfOutOfStocks+
                   RelatedCallsCount+OverallSubstitutionPolicy+Month+
                   (1|sequenceID),
                family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm1)

# use this line if predicting for existing customers
# pred <- predict(fitlrmm1, newdata=merged_customers_sc_test, allow.new.levels = TRUE,
# type="response")

# use this line if predicting for a new customer
pred <- predict(fitlrmm1, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
```

```r
tab <- table(merged_customers_sc_test$Churn, pred)
tab
# table elements:
# tab[1]      tab[3]
# tab[2]      tab[4]

# What proportion of dormant customers are missclassified?
tab[2]/(tab[2]+tab[4])

# What proportion of those who have stayed shopping are missclassified?
tab[3]/(tab[3]+tab[1])

# What proportion of the predicted leavers actually left?
tab[4]/(tab[4]+tab[3])

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# what percentage of the test observations are correctly classified?
sum(diag(tab))/sum(tab)
```

Model 2

```r
# random effect : sequenceID, StoreID
# fixed effects : all other predictors
fitlrmm2 <- glmer(Churn ~ ExpectedFulfillmentCharge+ActualCharge+TotalOrderItems+
                    TotalPickedLines+TotalQtySubbed+TotalQtyOOS+TotalPickTimeSeconds+
                    AvailabilityPreSubPercentage+PercentageOutOfStocks+
                    PercentageSubstitutions+ValueOfSubstitutions+ValueOfOutOfStocks+
                    RelatedCallsCount+OverallSubstitutionPolicy+Month+
                    (1|sequenceID)+(1|StoreID),
                 family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm2)

pred <- predict(fitlrmm2, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What proportion of dormant customers are missclassified?
tab[2]/(tab[2]+tab[4])

# What proportion of those who have stayed shopping are missclassified?
tab[3]/(tab[3]+tab[1])

# What proportion of the predicted leavers actually left?
tab[4]/(tab[4]+tab[3])

# What is the overall error rate for the test data?
```

```r
mean(pred != merged_customers_sc_test$Churn)

# what percentage of the test observations are correctly classified?
sum(diag(tab))/sum(tab)
```

Compare

```r
# compare models fitlrmm1 and fitlrmm2 to see if the random effect for StoreID is needed
# in the model
anova(fitlrmm1, fitlrmm2)
```

drop1 analysis and further models

```r
# Try the drop1 function - it compares the overall model (fitlrmm2) with the model
# resulting from removing that one specific variable. Choose a variable to remove
# based on a combination of low LRT/high p-val/high AIC
drop1(fitlrmm2, test="Chisq")
```

Model 3

```r
# TotalPickTimeSeconds, ValueOfSubstitutions removed
fitlrmm3 <- glmer(Churn ~ ExpectedFulfillmentCharge+ActualCharge+TotalOrderItems+
                   TotalPickedLines+TotalQtySubbed+TotalQtyOOS+
                   AvailabilityPreSubPercentage+
                   PercentageSubstitutions+PercentageOutOfStocks+ValueOfOutOfStocks+
                   RelatedCallsCount+OverallSubstitutionPolicy+Month+
                   (1|sequenceID)+(1|StoreID),
                family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm3)

pred <- predict(fitlrmm3, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, highest AIC, highest p-value
drop1(fitlrmm3, test="Chisq")
```

Model 4

```
# PercentageOutOfStocks, PercentageSubstitutions removed
fitlrmm4 <- glmer(Churn ~ ExpectedFulfillmentCharge+ActualCharge+TotalOrderItems+
                   TotalPickedLines+TotalQtySubbed+TotalQtyOOS+
                   AvailabilityPreSubPercentage+ValueOfOutOfStocks+
                   RelatedCallsCount+OverallSubstitutionPolicy+Month+
                   (1|sequenceID)+(1|StoreID),
                family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm4)

pred <- predict(fitlrmm4, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, highest AIC, highest p-value
drop1(fitlrmm4, test="Chisq")
```

Model 5

```
# ValueOfOutOfStocks removed
fitlrmm5 <- glmer(Churn ~ ExpectedFulfillmentCharge+ActualCharge+TotalOrderItems+
                   TotalPickedLines+TotalQtySubbed+TotalQtyOOS+
                   AvailabilityPreSubPercentage+RelatedCallsCount+
                   OverallSubstitutionPolicy+Month+
                   (1|sequenceID)+(1|StoreID),
                family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm5)

pred <- predict(fitlrmm5, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, highest AIC, highest p-value
drop1(fitlrmm5, test="Chisq")
```

Model 6

```r
# ExpectedFulfillmentCharge removed
fitlrmm6 <- glmer(Churn ~ ActualCharge+TotalOrderItems+TotalPickedLines+
                    TotalQtySubbed+TotalQtyOOS+
                   AvailabilityPreSubPercentage+RelatedCallsCount+
                   OverallSubstitutionPolicy+Month+
                   (1|sequenceID)+(1|StoreID),
                family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm6)

pred <- predict(fitlrmm6, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, highest AIC, highest p-value
drop1(fitlrmm6, test="Chisq")
```

Model 7

```r
# TotalOrderItems removed
fitlrmm7 <- glmer(Churn ~ ActualCharge+TotalPickedLines+
                    TotalQtySubbed+TotalQtyOOS+
                   AvailabilityPreSubPercentage+RelatedCallsCount+
                   OverallSubstitutionPolicy+Month+
                   (1|sequenceID)+(1|StoreID),
                family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm7)

pred <- predict(fitlrmm7, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, hightest AIC, highest p-value
drop1(fitlrmm7, test="Chisq")
```

Model 8

```r
# TotalQtyOOS removed
fitlrmm8 <- glmer(Churn ~ ActualCharge+TotalPickedLines+TotalQtySubbed+
                    AvailabilityPreSubPercentage+RelatedCallsCount+
                    OverallSubstitutionPolicy+Month+
                    (1|sequenceID)+(1|StoreID),
                 family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm8)

pred <- predict(fitlrmm8, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, hightest AIC, highest p-value
drop1(fitlrmm8, test="Chisq")
```

Model 9

```r
# RelatedCallsCount removed
fitlrmm9 <- glmer(Churn ~ ActualCharge+TotalPickedLines+TotalQtySubbed+
                    AvailabilityPreSubPercentage+OverallSubstitutionPolicy+
                    Month+(1|sequenceID)+(1|StoreID),
                 family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm9)

pred <- predict(fitlrmm9, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, hightest AIC, highest p-value
drop1(fitlrmm9, test="Chisq")
```

Model 10

```
# TotalQtySubbed removed
fitlrmm10 <- glmer(Churn ~ ActualCharge+TotalPickedLines+
                     AvailabilityPreSubPercentage+OverallSubstitutionPolicy+
                     Month+(1|sequenceID)+(1|StoreID),
                  family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm10)

pred <- predict(fitlrmm10, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, hightest AIC, highest p-value
drop1(fitlrmm10, test="Chisq")
```

Model 11

```
# AvailabilityPreSubPercentage removed
fitlrmm11 <- glmer(Churn ~ ActualCharge+TotalPickedLines+
                     OverallSubstitutionPolicy+Month+(1|sequenceID)+(1|StoreID),
                  family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm11)

pred <- predict(fitlrmm11, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, hightest AIC, highest p-value
drop1(fitlrmm11, test="Chisq")


# Create a model with an interaction between ActualCharge*Month
# These coefficients can be plotted
fitlrmm11a <- glmer(Churn ~ ActualCharge+TotalPickedLines+
                     OverallSubstitutionPolicy+Month+ActualCharge*Month+
```

```r
                    (1|sequenceID)+(1|StoreID),
               family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm11a)
summary(fitlrmm11a)$coefficients
class(summary(fitlrmm11a)$coefficients)
summary(fitlrmm11a)$coefficients[,1]
summary(fitlrmm11a)$coefficients[,2]

pred <- predict(fitlrmm11a, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)
```

```r
# ANOVA shows the interaction is not significant
anova(fitlrmm11, fitlrmm11a)

# Quantifying the interaction of ActualCharge*Month by looking at the coefficients.
df <- data.frame(Month=c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct",
                        "Nov","Dec"),
                ModelEst=c(-0.29748694, -0.27205429, -0.21857563, -0.45331214,
                           0.0000, -0.11188109, 0.15574785, -0.14187491,
                           -0.20965060, -0.26404848, -0.26297003, 0.08024445))

df$Month = factor(df$Month, levels = month.abb)

ggplot(df, aes(Month, ModelEst), y=ModelEst, fill=ModelEst) +
  geom_point(color = "green", size = 2) +
  ggtitle("Model Estimate Analysis for the predictor 'Month*ActualCharge'") +
  ylab("Model Estimate") + xlab("Month") +
  scale_x_discrete(limits = month.abb)
```

Model 12

```r
# consider interactions
fitlrmm12 <- glmer(Churn ~ ActualCharge+TotalPickedLines+
                   OverallSubstitutionPolicy+Month +
                   (ActualCharge+TotalPickedLines+OverallSubstitutionPolicy+Month)^2+
                   (1|sequenceID)+(1|StoreID),
               family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm12)

pred <- predict(fitlrmm12, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))
```

```r
# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, hightest AIC, highest p-value
drop1(fitlrmm12, test="Chisq")
```

Model 13

```r
# consider interaction between ActualCharge and TotalPickedLines
fitlrmm13 <- glmer(Churn ~ (ActualCharge*TotalPickedLines)+
                     ActualCharge+TotalPickedLines+
                     OverallSubstitutionPolicy+Month+
                     (1|sequenceID)+(1|StoreID),
                  family=binomial, nAGQ=0, data=merged_customers_sc_train)

summary(fitlrmm13)

pred <- predict(fitlrmm13, newdata=merged_customers_sc_test, re.form = NA, type="response")

pred <- factor(ifelse(pred < 0.5, 0, 1))

# generate a confusion matrix
tab <- table(merged_customers_sc_test$Churn, pred)
tab

# What is the overall error rate for the test data?
mean(pred != merged_customers_sc_test$Churn)

# Check to see which variable has the lowest LRT, hightest AIC, highest p-value
drop1(fitlrmm13, test="Chisq")
```

ANOVA

```r
# Are interactions significant
anova(fitlrmm11, fitlrmm12)
anova(fitlrmm11, fitlrmm13)
```

Analysis of “Month” Estimates

```r
# Extract the model coefficients
summary(fitlrmm11)$coefficients
class(summary(fitlrmm11)$coefficients)
# this will show the coefficient estimates
summary(fitlrmm11)$coefficients[,1]
# this will show the standard errors
```

```
summary(fitlrmm11)$coefficients[,2]

df <- data.frame(Month=c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct",
                         "Nov","Dec"),
                 ModelEst=c(0.9173198, 1.1433447, 1.1982748, 2.5004555, 0.000,
                            0.6168236, 0.6838760, 0.5270159, 0.6123175, 0.7038038,
                            0.8708587, 1.0187724),
                 StdErr=c(0.9707199, 0.9489250, 0.9375686, 0.8561449, 0.000,
                          1.1002202, 1.0992690, 1.1249372, 1.0638296, 1.0863399,
                          1.0152812, 1.0059335))

df$Month = factor(df$Month, levels = month.abb)

ggplot(df, aes(Month, ModelEst), y=ModelEst, fill=ModelEst) +
  geom_point(color = "red", size = 2) +
  geom_errorbar(aes(ymin=ModelEst-2*StdErr, ymax=ModelEst+2*StdErr), width=.2) +
  ggtitle("Model Estimate Analysis for the predictor 'Month'") +
  ylab("Model Estimate") + xlab("Month") +
  scale_x_discrete(limits = month.abb)

# How many customers churn for each of the months?
sum(merged_customers$Churn==1 & merged_customers$Month==1)
sum(merged_customers$Churn==1 & merged_customers$Month==2)
sum(merged_customers$Churn==1 & merged_customers$Month==3)
sum(merged_customers$Churn==1 & merged_customers$Month==4)
sum(merged_customers$Churn==1 & merged_customers$Month==5)
sum(merged_customers$Churn==1 & merged_customers$Month==6)
sum(merged_customers$Churn==1 & merged_customers$Month==7)
sum(merged_customers$Churn==1 & merged_customers$Month==8)
sum(merged_customers$Churn==1 & merged_customers$Month==9)
sum(merged_customers$Churn==1 & merged_customers$Month==10)
sum(merged_customers$Churn==1 & merged_customers$Month==11)
sum(merged_customers$Churn==1 & merged_customers$Month==12)
```

Plotting ActualCharge against Churn with a glm fitted per 'Month'

```
ggplot(merged_customers, aes(x = ActualCharge, y = log(as.numeric(Churn)), color = Month)) +
  scale_y_continuous(limits = c(-1.25, 1.0)) +
  ggtitle("ActualCharge plotted against Churn - fitting a glm per 'Month'") +
  ylab("logit(Churn)") +
  xlab("ActualCharge in Euro") +
  geom_smooth(method = glm, se = F)
```

Distribution of the Random Effects

```
# Look at the distribution of the customer random effects. Extract the random effects from
# the best model.
r_reducedpreds <- ranef(fitlrmm11)
re_reducedpreds <- r_reducedpreds$sequenceID[,1]
length(re_reducedpreds)
```

```r
plot(re_reducedpreds)
plot(1:1629, sort(re_reducedpreds))
hist(re_reducedpreds, main="Customer (sequenceID) random effect",
     xlab="Customer predicted random effects", col="darkcyan")

# Look at the distribution of the store random effects.
re_reducedpreds2 <- r_reducedpreds$StoreID[,1]
length(re_reducedpreds2)
plot(re_reducedpreds2)
plot(1:109, sort(re_reducedpreds2))
hist(re_reducedpreds2, main="StoreID random effect",
     xlab="Store predicted random effects", col="cornsilk2")
```