Prediction of specific game behavior

INTRODUCTION

Video games as virtual worlds have been believed to reflect human players' real world characteristics in diverse ways. For example, Yee et al. [1] correlated gameplay behavior with FFM personality model for the game World of Warcraft, finding significant correlations for personality traits. Van Lankveld et al. [2, 3] investigated correlations between real world personality traits and in-game behaviors via a modification of an existing game, Neverwinter Nights (Atari, 2002).

In this study, we aim to study whether it is possible to predict a specific game behavior in a Role Play Game, Fallout 3, based on players' real world characteristics (e.g. personality trait scores, game expertise and gender information). Precise and correct prediction of in-game actions will, for example, help game designers provide more customized and dynamic contents to players.

Our results showed that a certain game behavior can be predicted with 87% prediction accuracy using a verified and robust logistic regression model. It is tested that the game behavior has significant association with several real world characteristics.

SUMMARY OF THE DATA:

We asked 64 subjects to play the game for one hour and had them take personality tests in a psychology lab at Northeastern University. Personality scores of the big five traits (Openness, Conscientiousness, Extroversion, Agreeableness and Neuroticism) were gathered in a standard way as suggested by [4]. The players were also asked to provide their gender (male/female) and game expertise (gamer/non-gamer) information.

The 64 game logs contained a large amount of in-game behaviors. A specific game behavior is extracted for study in this project. We denote this behavior *Seq0* in the following text. The game behavior is characterized by talking to a Non-Person Character (NPC) three times during gameplay.

METHODS

We used the big five trait scores (numeric), gender information (binary, male/female) and game expertise (binary, gamer/non-gamer) of players as the **predictor variables**. In total there are seven individual predictor variables. Moreover, we also considered to use second-order interaction terms as the predictor variables. The seven individual predictor variables have 7*6/2=21 second-order interaction terms in total. Whether a player performed Seq0 in his gameplay was set as the **response variable**. Since our response variable is binary, we are going for Logistic Regression to make an analysis. Steps to be followed are:

Data normalization: Every personality trait predictor variable is scaled to have mean 0 and standard deviation 1. The normalization step will help interpret our model coefficients more easily.

Understanding Data: We initially understand the data by plotting pairwise scatterplots. From the pairwise scatterplots we can know the data distribution as well as examine the correlations between the predictors.

Likelihood ratio test: We build a *full model* with all individual predictors as well as interaction variables, and a *reduced model* with no interaction variables. We perform the likelihood ratio test to see which model is more appropriate to explain the response variable. We use

$$H_0: \beta_i X_i \dots\dots \beta_p X_p = 0$$
$$H_a: \beta_i X_i \dots\dots \beta_p X_p \neq 0$$

The actual test statistic for the likelihood ratio test, denoted by G_2 is:

$$G_2 = -2\log_e L(R)/L(F) = -2[\log_e L(R) - \log_e L(F)]$$

Large-sample theory states that when data size is large, G_2 is distributed approximately as $\chi_2(i-1)$ when H_0 holds. The approximate decision rule therefore is:

If $G2 \leq \chi2(1-\alpha;p-q)$, conclude $H0$

If $G2> \chi2(1-\alpha;p-q)$, conclude $H\alpha$

After the Likelihood test, we find the full model is appropriate one.

Variable Selection Method: We performed three variable selection methods over the full model. Every method is applied on the dataset with 10-fold cross validation. A final model is selected with the best prediction accuracy on average in the cross validation.

- **Lasso method:** The lasso method adds a penalty term in the least square objective function in the training, often resulting in less model variance but higher bias. The penalty term is controlled by the parameter λ. In the Lasso method, we perform a grid search on different values of λ and verify them through the cross validation.
- **Stepwise AIC:** We perform a stepwise variable selection method with bi-directions based on the *Akaike Information Criterion* (AIC) metric. A model with a low test error will have a small value for AIC.
- **Stepwise BIC:** We perform a stepwise variable selection method with bi-directions based on the Bayesian Information Criterion (BIC) metric. A model with a low test error will have a small value for BIC.

Eventually, the lasso method returns a model with proper λ and the best prediction accuracy in the cross validation. Only four predictor variables are associated with non-zero weights, namely, “Gamer”, “C:A”, “E:N” and “E:Sex”. (The last three are interaction variables.)

The proceeding tests are based on the model returned by the lasso method. The notation for the model is as follows:

$$\log\left(\frac{E\{Y_i\}}{1-E\{Y_i\}}\right) = \beta_0 + \beta_1 * Gamer + \beta_2 * C * A + \beta_3 * E * N + \beta_4 * E * Sex$$

Test for Goodness of fit: We used Hosmer and Lemeshow test to see whether the fit we derived is a good fit.

$$H_0 : \log\left(\frac{E\{Y_i\}}{1-E\{Y_i\}}\right) = \beta_0 + \beta_1 * Gamer + \beta_2 * C * A + \beta_3 * E * N + \beta_4 * E * Sex$$

$$H_a : \log\left(\frac{E\{Y_i\}}{1-E\{Y_i\}}\right) \neq \beta_0 + \beta_1 * Gamer + \beta_2 * C * A + \beta_3 * E * N + \beta_4 * E * Sex$$

The conclusion is that since the p-value is high, we failed to reject the null hypothesis.

Residual Diagnostics: We also perform the residual diagnostics such as plot on fitted values vs residuals values, Normal Q-Q plot, Deviance Residuals plot, Cook’s distance to find the influential observation and cases. From the four plots, we identified some outliers that might affect the model estimation. Hence we removed the respective data.

ROC: We now use the roc curve to measure the predictive ability of the model using the cross validation K=10 on the dataset. The average area under the curve is 0.87.

Visualizing predicted probability: Finally, we plot the predicted probabilities as a function of one predictor variable while other predictor variables are constant.

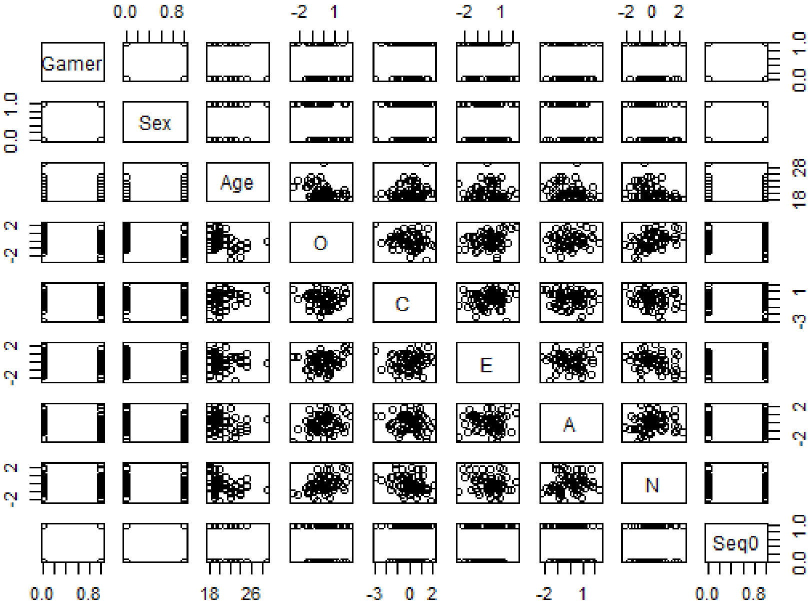
Association Discovery: we would like to see whether any single predictor variable is associated with the execution of Seq0. This can be interpreted from the 95% confidence intervals for the coefficient estimates. For a predictor variable, if the entire 95% confidence interval for its associated coefficient estimate is above 1, we can conclude positive association. If the entire 95% confidence interval for its coefficient estimate is below 1, we can conclude negative association. Otherwise we cannot conclude there is an association.

One thousand bootstrap experiments are used to obtain the confidence intervals for the coefficient estimates. In each bootstrap, we sample 64 data points from the original dataset with replacement, on which a full model of logistic regression (with L1 regularization) is trained and the coefficient estimates are returned. After the one thousand bootstraps, we have a bootstrap distribution for each coefficient and its 5% and 95% quantiles can be identified to get the confidence intervals.

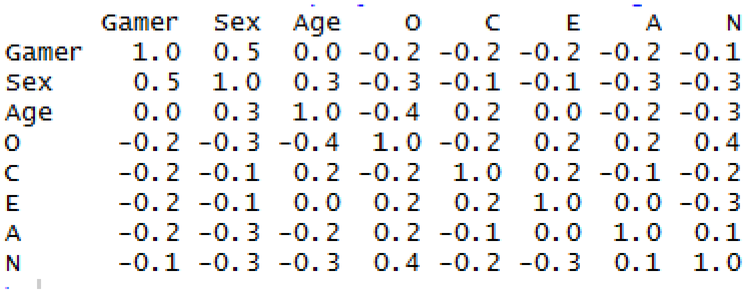
RESULTS

Understanding the Data

The pairwise scatterplot shown below helps us to understand the dataset. There are no evident correlations to be identified.



We also see the results of the correlation matrix. Most of the correlations are weak or mild. So we don’t take further remedial step.

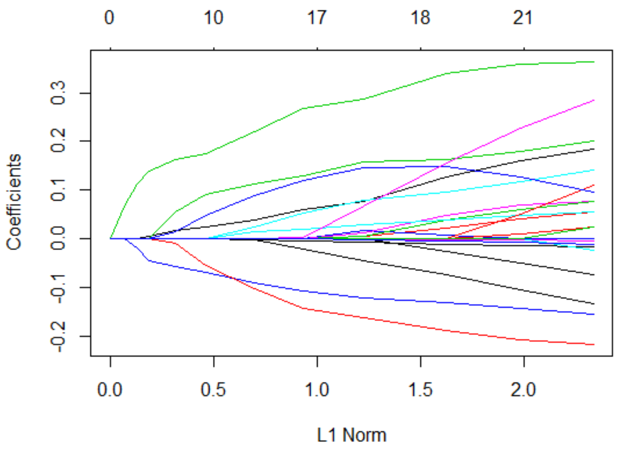


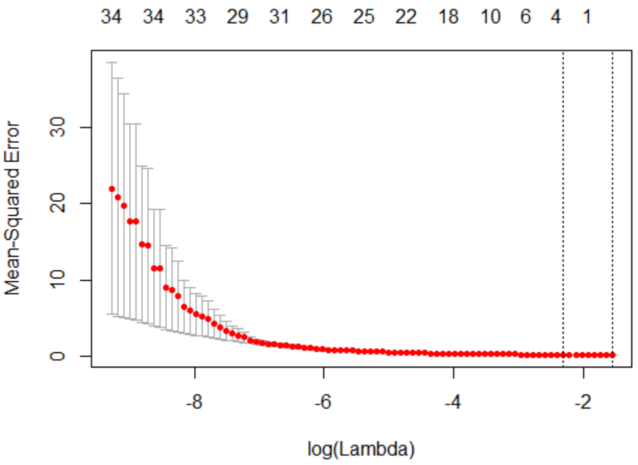
```
        Gamer  Sex  Age    O    C    E    A    N
Gamer     1.0  0.5  0.0 -0.2 -0.2 -0.2 -0.2 -0.1
Sex       0.5  1.0  0.3 -0.3 -0.1 -0.1 -0.3 -0.3
Age       0.0  0.3  1.0 -0.4  0.2  0.0 -0.2 -0.3
O        -0.2 -0.3 -0.4  1.0 -0.2  0.2  0.2  0.4
C        -0.2 -0.1  0.2 -0.2  1.0  0.2 -0.1 -0.2
E        -0.2 -0.1  0.0  0.2  0.2  1.0  0.0 -0.3
A        -0.2 -0.3 -0.2  0.2 -0.1  0.0  1.0  0.1
N        -0.1 -0.3 -0.3  0.4 -0.2 -0.3  0.1  1.0
```

Variable selection Method

Lasso Method

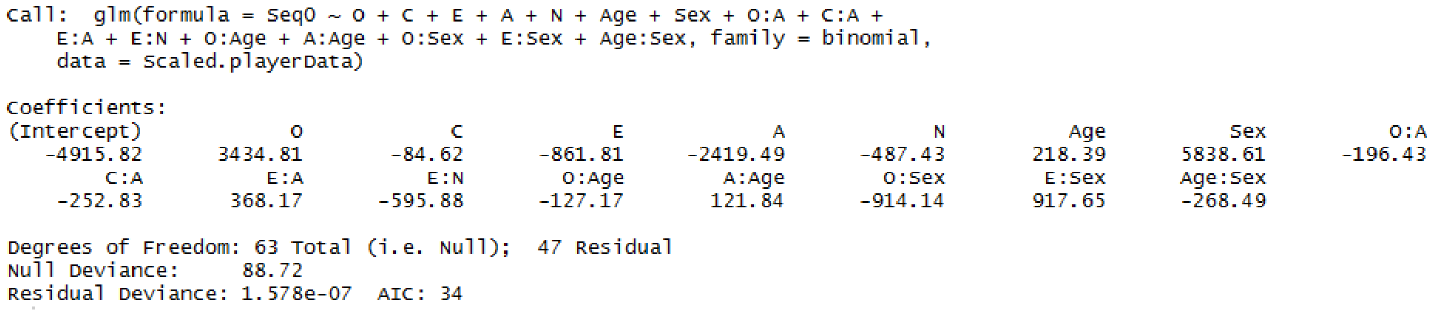
The lasso plot is as shown below. We can see that depending on the choice of the tuning parameter, some of the coefficients are exactly equal to zero. We perform the cross-validation and compute the associated test error. The cross-validation plot is as shown below:





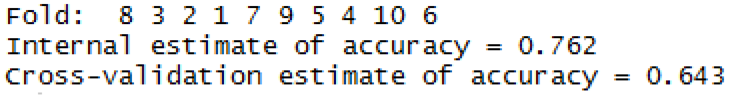
Step AIC method and Step BIC method

We received 15 variables from both the methods. The result is as shown below.

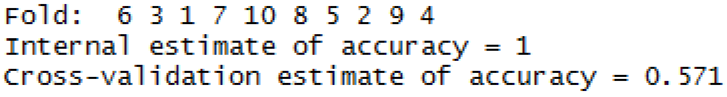

```
Call:  glm(formula = Seq0 ~ O + C + E + A + N + Age + Sex + O:A + C:A + 
    E:A + E:N + O:Age + A:Age + O:Sex + E:Sex + Age:Sex, family = binomial, 
    data = Scaled.playerData)

Coefficients:
(Intercept)            O            C            E            A            N          Age          Sex          O:A  
   -4915.82      3434.81       -84.62      -861.81     -2419.49      -487.43       218.39      5838.61      -196.43  
        C:A          E:A          E:N        O:Age        A:Age        O:Sex        E:Sex      Age:Sex  
    -252.83       368.17      -595.88      -127.17       121.84      -914.14       917.65      -268.49  

Degrees of Freedom: 63 Total (i.e. Null);  47 Residual
Null Deviance:	    88.72 
Residual Deviance: 1.578e-07 	AIC: 34
```

Finally in order to select the best model selection method, we preform cross-validation for the 3 methods.

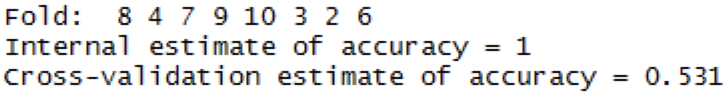
- Lasso Method:


```
Fold:  8 3 2 1 7 9 5 4 10 6
Internal estimate of accuracy = 0.762
Cross-validation estimate of accuracy = 0.643
```

- StepAIC Method:


```
Fold:  6 3 1 7 10 8 5 2 9 4
Internal estimate of accuracy = 1
Cross-validation estimate of accuracy = 0.571
```

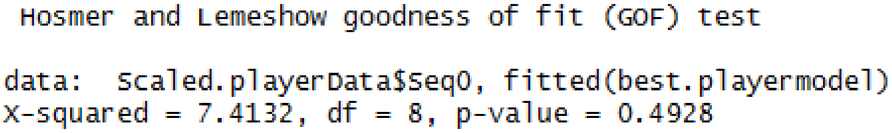
- StepBIC Method:


```
Fold:  8 4 7 9 10 3 2 6
Internal estimate of accuracy = 1
Cross-validation estimate of accuracy = 0.531
```

From the results above, we see that the lasso method gives the best predictor variables such as Gamer, C:A, E:N, E:Sex. The best lambda value is 0.1002417.

Test for goodness of fit:

The results of Hosmer and Lemeshow test is as shown below.



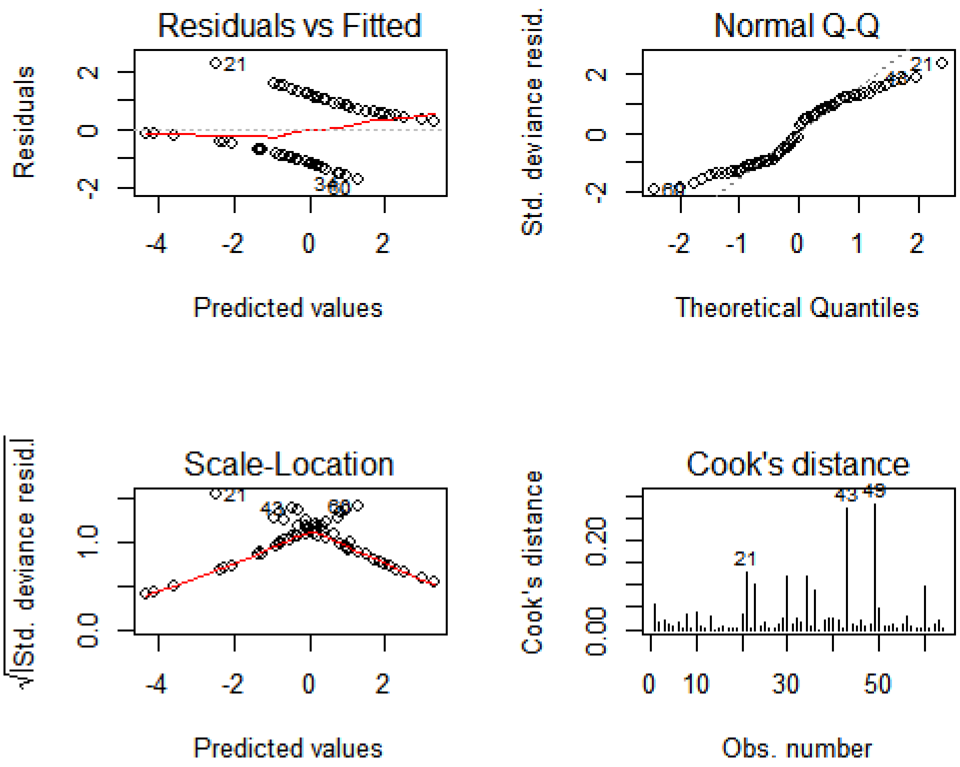
```
 Hosmer and Lemeshow goodness of fit (GOF) test

data:  Scaled.playerData$Seq0, fitted(best.playermodel)
X-squared = 7.4132, df = 8, p-value = 0.4928
```

The p-value is high, hence we fail to reject the null hypothesis.

Residual diagnostics

We performed the 4 diagnostics on the residuals of the model.

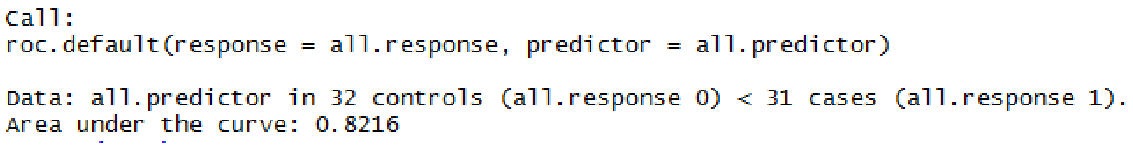
- **Residuals vs Fitted:** The plot below suggest that if the model is correct, a lowess smooth of the plot of the residuals against the estimated probability should result approximately in a horizontal line with zero intercept. So this shows that the model is correct, except the observation 21 as potential outliers.
- **Normal Q-Q:** In this plot, the observation 21 seem to be the outliers.
- **Scale-Location:** The predicted values are plotted against the absolute value of $\sqrt{|Std\ deviance\ resid|}$. Since these are not signed residuals, we do not expect horizontal smooth line. Here also, we see that the observation 21 and 43 are potential outliers or influential points.
- **Cook's distance:** This measures the summary change in predicted values after removing each individual observation. The observation 21, 43 and 49 were identified as influential observations.



In combination of the four plots, we determined to remove the data point #21 from our dataset since we find that it is an outlier.

ROC:

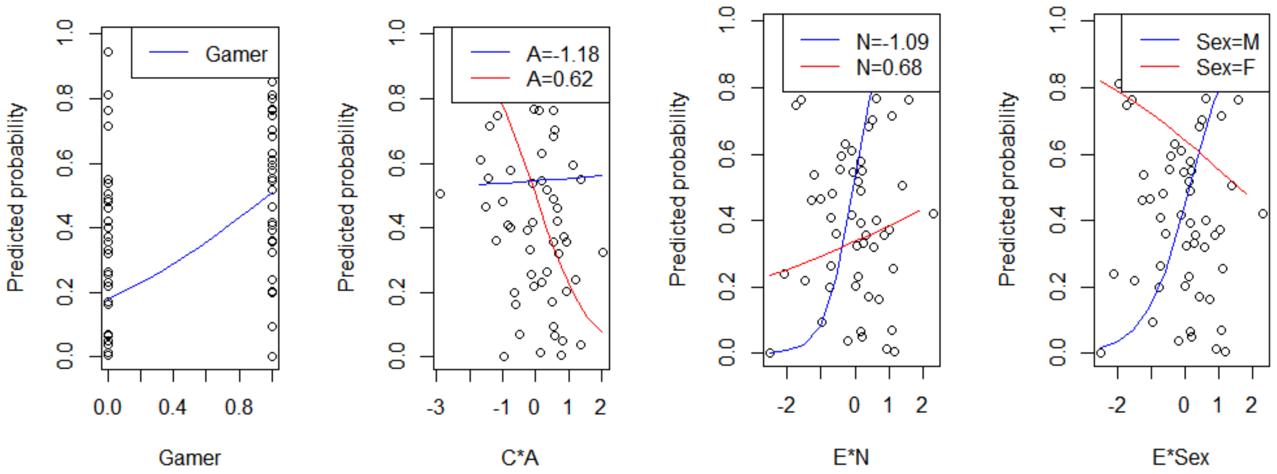
We perform cross-validation for ROC and the results are as shown below:



```
Call:
roc.default(response = all.response, predictor = all.predictor)

Data: all.predictor in 32 controls (all.response 0) < 31 cases (all.response 1).
Area under the curve: 0.8216
```

The mean of all the AUC is 0.879.

Visualizing predicted probability:



Plot 1: From the plot, we conclude that Gamer has a positive impact on the predicted probability of Seq0, i.e., when the player is a gamer there is more chance of performing the behavior Seq0.

Plot 2: From the plot, we conclude the effect of Conscientiousness on Seq0 at different values of Agreeableness:

- When Agreeable score is mean (A) + std(A)=0.62, then the predicted probability is decreases, i.e., there is less chance of the player to perform Seq0.
- When Agreeable score is mean (A) – std(A)=-1.18, then the predicted probability is constant, i.e., there is same chance of the player to perform the sequence of actions.

Plot 3: From the plot, we can see the effect of Extroversion on Seq0 at different values of Neuroticism. When Neuroticism score is either mean (N) + std(N)=0.68 or mean(N) - std(N)=-1.09, the predicted probability is increasing, i.e., there is more chance of a player to perform Seq0 if the player is more extrovert.

Plot 4: From the plot, when the player is Male and extrovert, then the player has more chance of performing the sequence of actions and when the player is female and extrovert, then the player has less chance of performing the sequence of actions.

DISCUSSIONS

Since the dataset size is limited, the additional step we would like to do is to collect more data. When the data is more sufficient, we would like to try other prediction algorithms, such as boosting.

REFERENCES

[1] Yee, N., Ducheneaut, N., Nelson, L., & Likarish, P. 2011, May. Introverted elves & conscientious gnomes: the expression of personality in world of warcraft. In Proceedings of the 2011 annual conference on Human factors in computing systems, 753-762.

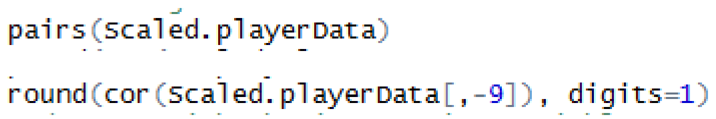
[2] Lankveld, G., Schreurs, S., Spronck, P., and van den Herik J. 2011b. Extraversion in Games. Computers and Games, 7th International Conference, CG2010 (eds. H. Jaap van den Herik, Hiroyuki Iida, and Aske Plaat), pp. 263-275. LNCS 6515. Springer-Verlag, Germany.

[3] Lankveld, G., Schreurs, S., Spronck, P. 2009. Psychologically Verified Player Modelling. 10th International Conference on Intelligent Games and Simulation GAME-ON 2009 (ed. Linda Breitlauch), pp. 12-19. EUROSIS.

[4] Gomariz, A., Campos, M., Marín, R., & Goethals, B. (2013). Clasp: An efficient algorithm for mining frequent closed sequences. In Advances in Knowledge Discovery and Data Mining (pp. 50-61). Springer Berlin Heidelberg.

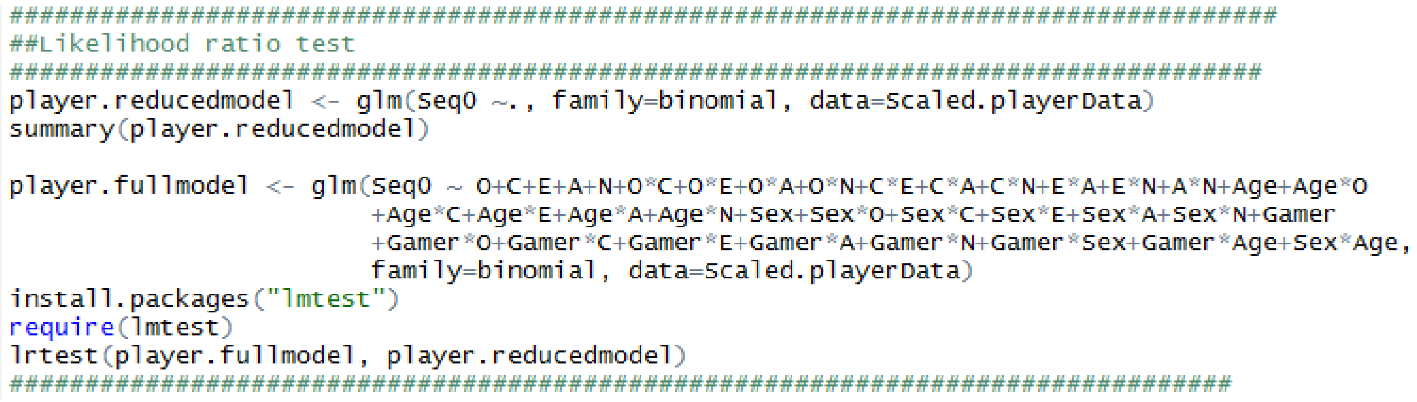
APPENDIX

Code for displaying pairs and correlation matrix:



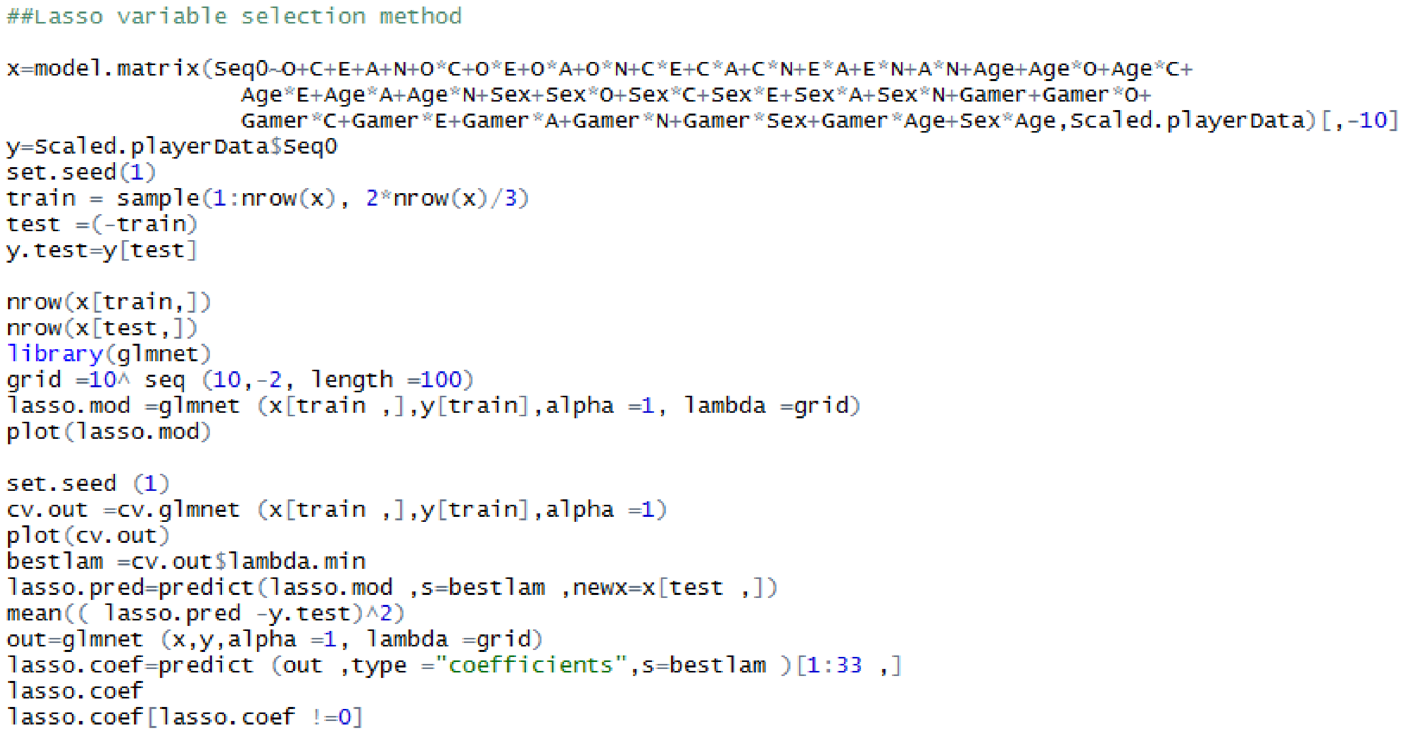
```
pairs(Scaled.playerData)

round(cor(Scaled.playerData[,-9]), digits=1)
```

Code for likelihood ratio test:

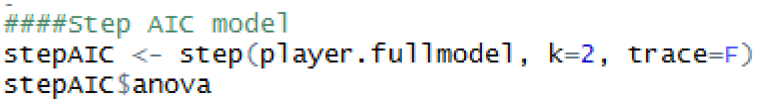


```
####################################################################################
##Likelihood ratio test
###################################################################################
player.reducedmodel <- glm(Seq0 ~., family=binomial, data=Scaled.playerData)
summary(player.reducedmodel)

player.fullmodel <- glm(Seq0 ~ O+C+E+A+N+O*C+O*E+O*A+O*N+C*E+C*A+C*N+E*A+E*N+A*N+Age+Age*O
                        +Age*C+Age*E+Age*A+Age*N+Sex+Sex*O+Sex*C+Sex*E+Sex*A+Sex*N+Gamer
                        +Gamer*O+Gamer*C+Gamer*E+Gamer*A+Gamer*N+Gamer*Sex+Gamer*Age+Sex*Age,
                        family=binomial, data=Scaled.playerData)
install.packages("lmtest")
require(lmtest)
lrtest(player.fullmodel, player.reducedmodel)
#################################################################################
```

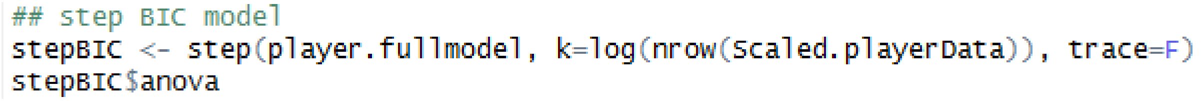
Code for variable selection method such as Lasso model, stepAIC and stepBIC:



```
##Lasso variable selection method

x=model.matrix(Seq0~O+C+E+A+N+O*C+O*E+O*A+O*N+C*E+C*A+C*N+E*A+E*N+A*N+Age+Age*O+Age*C+
                 Age*E+Age*A+Age*N+Sex+Sex*O+Sex*C+Sex*E+Sex*A+Sex*N+Gamer+Gamer*O+
                 Gamer*C+Gamer*E+Gamer*A+Gamer*N+Gamer*Sex+Gamer*Age+Sex*Age,Scaled.playerData)[,-10]
y=Scaled.playerData$Seq0
set.seed(1)
train = sample(1:nrow(x), 2*nrow(x)/3)
test =(-train)
y.test=y[test]

nrow(x[train,])
nrow(x[test,])
library(glmnet)
grid =10^ seq (10,-2, length =100)
lasso.mod =glmnet (x[train ,],y[train],alpha =1, lambda =grid)
plot(lasso.mod)

set.seed (1)
cv.out =cv.glmnet (x[train ,],y[train],alpha =1)
plot(cv.out)
bestlam =cv.out$lambda.min
lasso.pred=predict(lasso.mod ,s=bestlam ,newx=x[test ,])
mean(( lasso.pred -y.test)^2)
out=glmnet (x,y,alpha =1, lambda =grid)
lasso.coef=predict (out ,type ="coefficients",s=bestlam )[1:33 ,]
lasso.coef
lasso.coef[lasso.coef !=0]
```

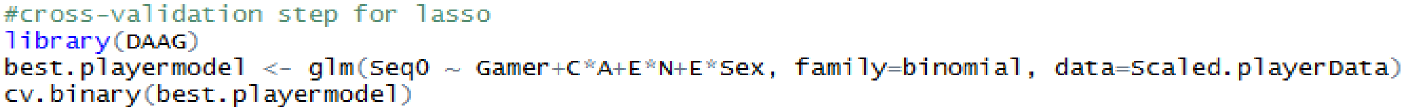


```
####Step AIC model
stepAIC <- step(player.fullmodel, k=2, trace=F)
stepAIC$anova
```

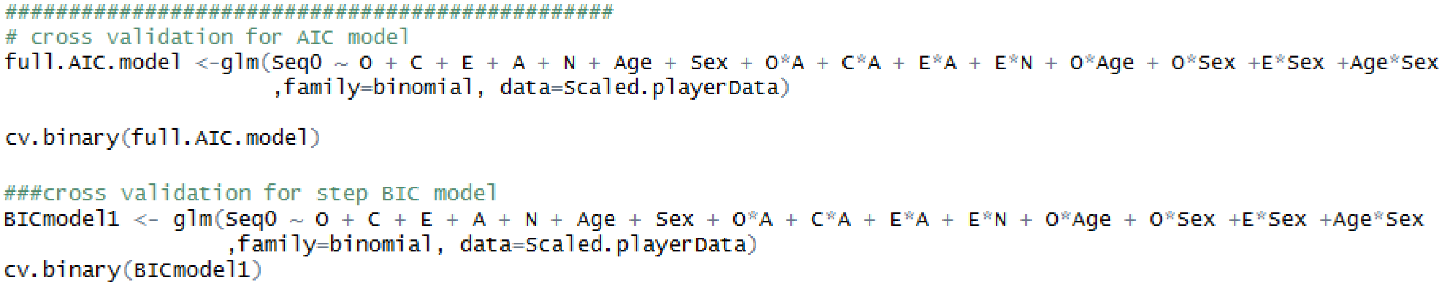


```
## step BIC model
stepBIC <- step(player.fullmodel, k=log(nrow(Scaled.playerData)), trace=F)
stepBIC$anova
```

Code for model selection using cross validation:

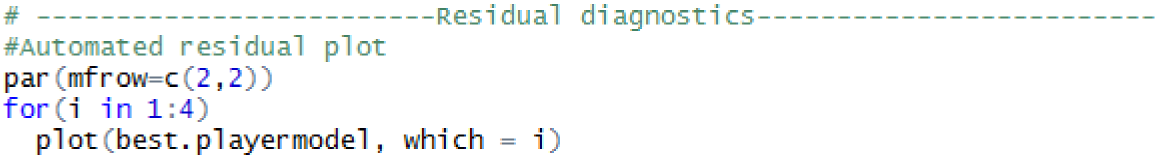


```
#cross-validation step for lasso
library(DAAG)
best.playermodel <- glm(Seq0 ~ Gamer+C*A+E*N+E*Sex, family=binomial, data=Scaled.playerData)
cv.binary(best.playermodel)
```



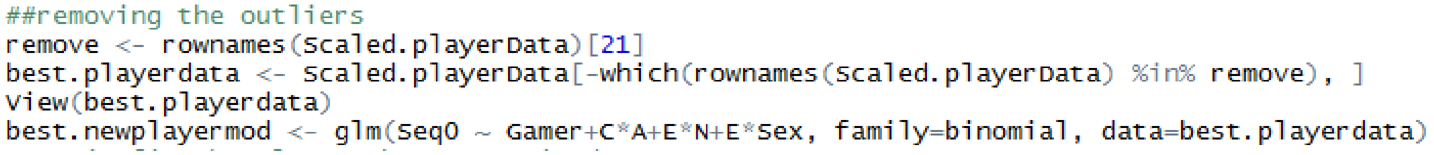
```
################################################
# cross validation for AIC model
full.AIC.model <-glm(Seq0 ~ O + C + E + A + N + Age + Sex + O*A + C*A + E*A + E*N + O*Age + O*Sex +E*Sex +Age*Sex
                     ,family=binomial, data=Scaled.playerData)

cv.binary(full.AIC.model)

###cross validation for step BIC model
BICmodel1 <- glm(Seq0 ~ O + C + E + A + N + Age + Sex + O*A + C*A + E*A + E*N + O*Age + O*Sex +E*Sex +Age*Sex
                 ,family=binomial, data=Scaled.playerData)
cv.binary(BICmodel1)
```

Code for residual plots:



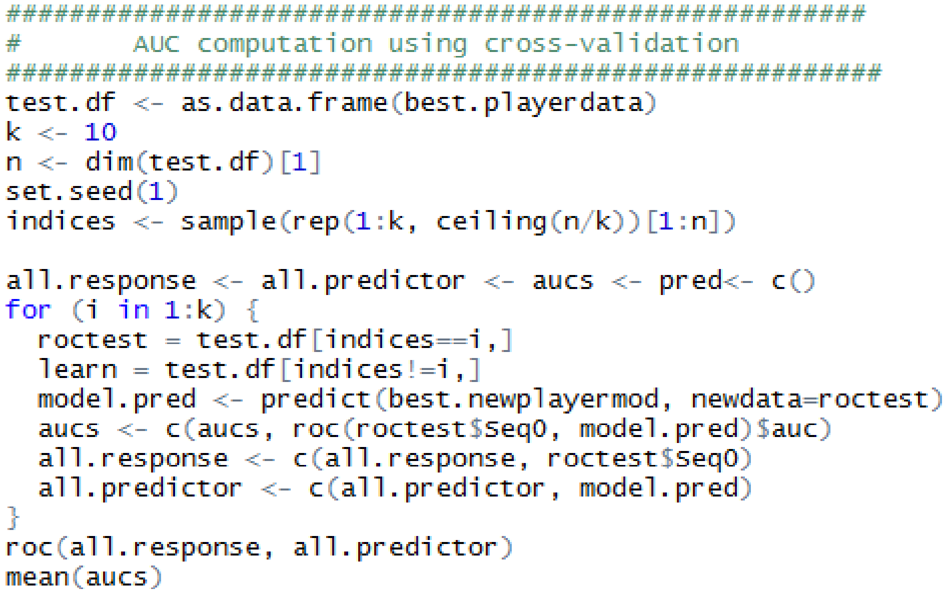
```
# -------------------------Residual diagnostics-------------------------
#Automated residual plot
par(mfrow=c(2,2))
for(i in 1:4)
  plot(best.playermodel, which = i)
```

Code for removing the outliers:



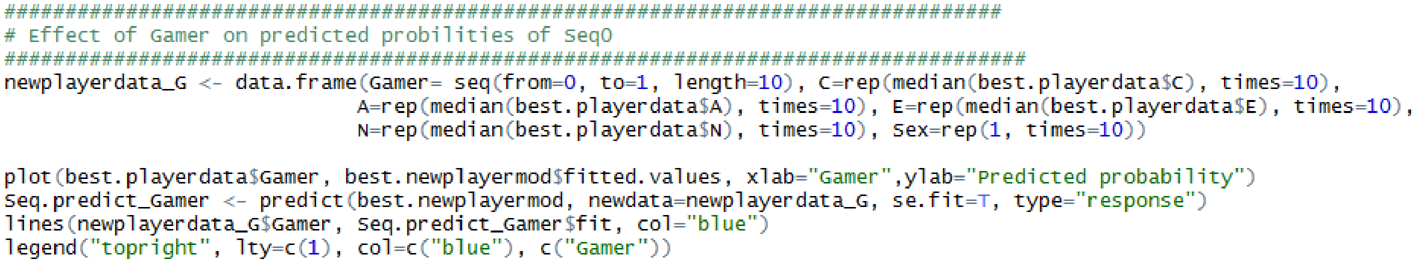
```
##removing the outliers
remove <- rownames(Scaled.playerData)[21]
best.playerdata <- Scaled.playerData[-which(rownames(Scaled.playerData) %in% remove), ]
View(best.playerdata)
best.newplayermod <- glm(Seq0 ~ Gamer+C*A+E*N+E*Sex, family=binomial, data=best.playerdata)
```

Code for ROC curves by performing cross validation K=10

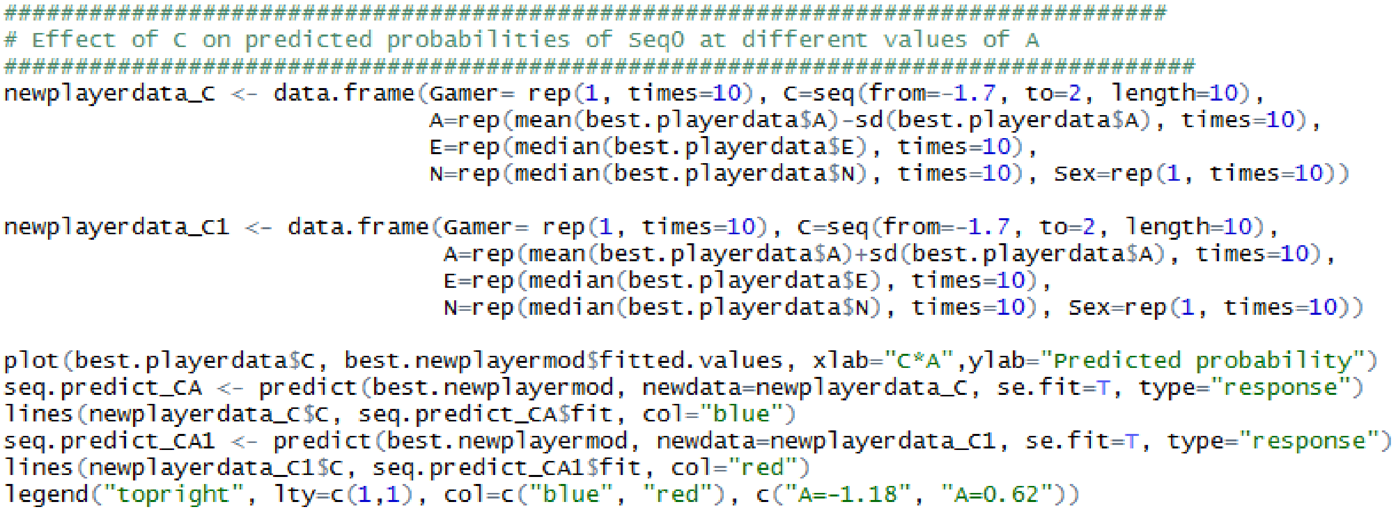


```
######################################################
#       AUC computation using cross-validation
#######################################################
test.df <- as.data.frame(best.playerdata)
k <- 10
n <- dim(test.df)[1]
set.seed(1)
indices <- sample(rep(1:k, ceiling(n/k))[1:n])

all.response <- all.predictor <- aucs <- pred<- c()
for (i in 1:k) {
  roctest = test.df[indices==i,]
  learn = test.df[indices!=i,]
  model.pred <- predict(best.newplayermod, newdata=roctest)
  aucs <- c(aucs, roc(roctest$Seq0, model.pred)$auc)
  all.response <- c(all.response, roctest$Seq0)
  all.predictor <- c(all.predictor, model.pred)
}
roc(all.response, all.predictor)
mean(aucs)
```

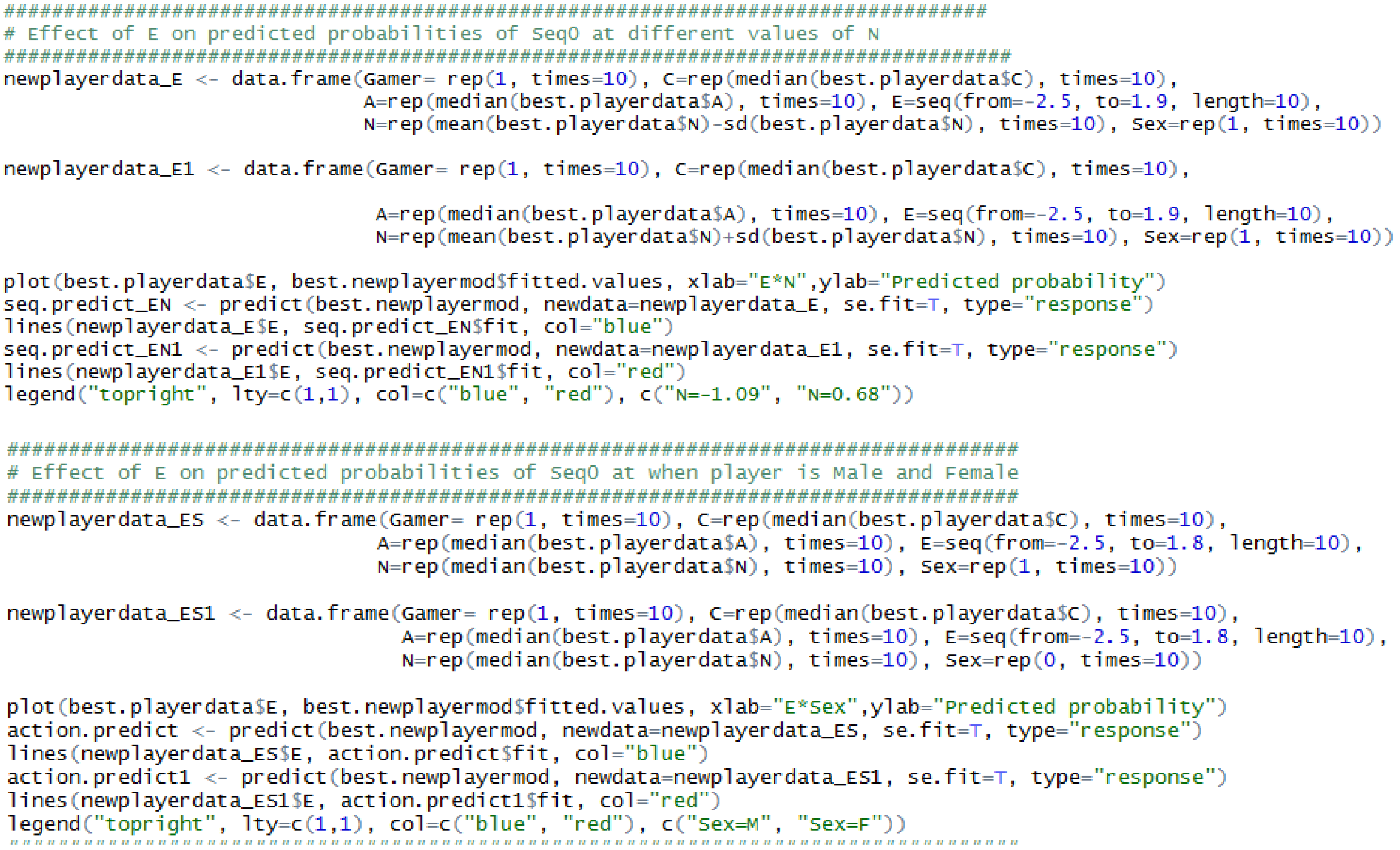
Code for the visualizations of predicted probabilities and the predictor variables



```
###################################################################################
# Effect of Gamer on predicted probilities of Seq0
#####################################################################################
newplayerdata_G <- data.frame(Gamer= seq(from=0, to=1, length=10), C=rep(median(best.playerdata$C), times=10),
                              A=rep(median(best.playerdata$A), times=10), E=rep(median(best.playerdata$E), times=10),
                              N=rep(median(best.playerdata$N), times=10), Sex=rep(1, times=10))

plot(best.playerdata$Gamer, best.newplayermod$fitted.values, xlab="Gamer",ylab="Predicted probability")
Seq.predict_Gamer <- predict(best.newplayermod, newdata=newplayerdata_G, se.fit=T, type="response")
lines(newplayerdata_G$Gamer, Seq.predict_Gamer$fit, col="blue")
legend("topright", lty=c(1), col=c("blue"), c("Gamer"))
```



```
###################################################################################
# Effect of C on predicted probabilities of Seq0 at different values of A
#####################################################################################
newplayerdata_C <- data.frame(Gamer= rep(1, times=10), C=seq(from=-1.7, to=2, length=10),
                              A=rep(mean(best.playerdata$A)-sd(best.playerdata$A), times=10),
                              E=rep(median(best.playerdata$E), times=10),
                              N=rep(median(best.playerdata$N), times=10), Sex=rep(1, times=10))

newplayerdata_C1 <- data.frame(Gamer= rep(1, times=10), C=seq(from=-1.7, to=2, length=10),
                               A=rep(mean(best.playerdata$A)+sd(best.playerdata$A), times=10),
                               E=rep(median(best.playerdata$E), times=10),
                               N=rep(median(best.playerdata$N), times=10), Sex=rep(1, times=10))

plot(best.playerdata$C, best.newplayermod$fitted.values, xlab="C*A",ylab="Predicted probability")
seq.predict_CA <- predict(best.newplayermod, newdata=newplayerdata_C, se.fit=T, type="response")
lines(newplayerdata_C$C, seq.predict_CA$fit, col="blue")
seq.predict_CA1 <- predict(best.newplayermod, newdata=newplayerdata_C1, se.fit=T, type="response")
lines(newplayerdata_C1$C, seq.predict_CA1$fit, col="red")
legend("topright", lty=c(1,1), col=c("blue", "red"), c("A=-1.18", "A=0.62"))
```



```r
###################################################################################
# Effect of E on predicted probabilities of Seq0 at different values of N
####################################################################################
newplayerdata_E <- data.frame(Gamer= rep(1, times=10), C=rep(median(best.playerdata$C), times=10),
                              A=rep(median(best.playerdata$A), times=10), E=seq(from=-2.5, to=1.9, length=10),
                              N=rep(mean(best.playerdata$N)-sd(best.playerdata$N), times=10), Sex=rep(1, times=10))

newplayerdata_E1 <- data.frame(Gamer= rep(1, times=10), C=rep(median(best.playerdata$C), times=10),

                               A=rep(median(best.playerdata$A), times=10), E=seq(from=-2.5, to=1.9, length=10),
                               N=rep(mean(best.playerdata$N)+sd(best.playerdata$N), times=10), Sex=rep(1, times=10))

plot(best.playerdata$E, best.newplayermod$fitted.values, xlab="E*N",ylab="Predicted probability")
seq.predict_EN <- predict(best.newplayermod, newdata=newplayerdata_E, se.fit=T, type="response")
lines(newplayerdata_E$E, seq.predict_EN$fit, col="blue")
seq.predict_EN1 <- predict(best.newplayermod, newdata=newplayerdata_E1, se.fit=T, type="response")
lines(newplayerdata_E1$E, seq.predict_EN1$fit, col="red")
legend("topright", lty=c(1,1), col=c("blue", "red"), c("N=-1.09", "N=0.68"))

####################################################################################
# Effect of E on predicted probabilities of Seq0 at when player is Male and Female
####################################################################################
newplayerdata_ES <- data.frame(Gamer= rep(1, times=10), C=rep(median(best.playerdata$C), times=10),
                               A=rep(median(best.playerdata$A), times=10), E=seq(from=-2.5, to=1.8, length=10),
                               N=rep(median(best.playerdata$N), times=10), Sex=rep(1, times=10))

newplayerdata_ES1 <- data.frame(Gamer= rep(1, times=10), C=rep(median(best.playerdata$C), times=10),
                                A=rep(median(best.playerdata$A), times=10), E=seq(from=-2.5, to=1.8, length=10),
                                N=rep(median(best.playerdata$N), times=10), Sex=rep(0, times=10))

plot(best.playerdata$E, best.newplayermod$fitted.values, xlab="E*Sex",ylab="Predicted probability")
action.predict <- predict(best.newplayermod, newdata=newplayerdata_ES, se.fit=T, type="response")
lines(newplayerdata_ES$E, action.predict$fit, col="blue")
action.predict1 <- predict(best.newplayermod, newdata=newplayerdata_ES1, se.fit=T, type="response")
lines(newplayerdata_ES1$E, action.predict1$fit, col="red")
legend("topright", lty=c(1,1), col=c("blue", "red"), c("Sex=M", "Sex=F"))
```