TheFinalProject

2023-04-26

Import libraries and connect to SQL

```
library(tidyverse)

library(tidymodels)

library(dplyr)
library(DBI)
library(odbc)
library(fastDummies)

library(boot)
library(plyr)

library(neuralnet)

library(caret)

library(gridExtra)

library(reshape2)

library(TTR)

library(randomForest)

# conn <- DBI::dbConnect(
#   odbc::odbc(),
#   Driver="SQL Server",
#   Server="COMPUTER_NAME",
#   Database="Your Database Name",
#   options(connectionObserver = NULL)
# )

norm = function(x){
  m0 = min(x)
  m1 = max(x)
  result = (x - m0)/(m1 - m0)
  return(result)
}
```

```
regular = function(x, y){
  m0 = min(y)
  m1 = max(y)
  result = x*(m1 - m0) + m0
  return(result)
}
```

Machine Learning Round One

Fetch datasets from SQL

```
# bike_donations <- dbGetQuery(conn, "SELECT TOP 50000 * FROM
dbo.BikeDonations")
# bike_events <- dbGetQuery(conn, "SELECT * FROM dbo.BikeEvents")
#write_csv(bike_donations, "BikeDonations.csv")
#write_csv(bike_events, "BikeEvents.csv")
```

Read generated csv files

```
bike_donations <- read_csv("BikeDonations.csv")

bike_events <- read_csv("BikeEvents.csv")
```

Join the tables and omit N/A variables

```
df2 <- left_join(x=bike_donations, y=bike_events, by="EventID")
df2[df2 == "N/A"] <- NA
df2 <- df2 %>% na.omit(df2)
```

Convert string variables from dataset into numeric

```
df3 <- df2 %>%
  mutate(GiftAmount=as.numeric(GiftAmt.x),
         Goals=as.numeric(Goals),
         ActiveReg=as.numeric(ActiveReg),
         NoReg=as.numeric(TotalFees),
         SentEmails=as.numeric(SentEmails)) %>%
  select(-EventID,-FiscalYear.x,-GiftAmt.x,-GiftAmt.y,-TotalFees,-
ConfirmedGifts,-TotalOnlineGifts,-FiscalYear.y,-CampID,-DonorConsID,-
Goals,-TeamID)
```

Fix some of the variables spacing and such

```
df3[df3 == "I have a Friend or Co-worker with MS"] <-
"FriendOrCoWorker"
```

```
df3[df3 == "Bad (Soft Bounce)"] <- "SoftBounce"
df3[df3 == "Bad (Hard Bounce)"] <- "HardBounce"
df3[df3 == "Relative: Parent of person with MS"] <- "RelativeParent"
df3[df3 == "Relative: Other"] <- "RelativeOther"
df3[df3 == "I have a Friend of Co-worker with MS"] <-
"FriendOfCoWorker"
```

Parse dummy variables in dataset

```
dataset <- fastDummies::dummy_cols(df3) %>%
  select(-GiftType,-PmtMethod,-Registered,-EmailStatus,-Connection)
colnames(dataset) = gsub(" ", "_", colnames(dataset))


pre_norm_set <- dataset
```

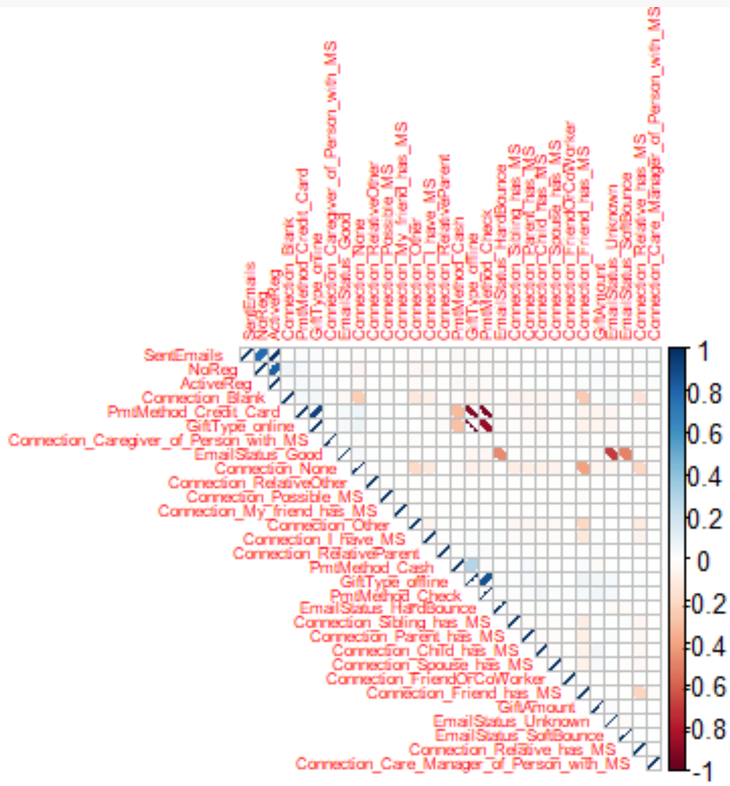
Extract Column Names

```
colnames(dataset)

##  [1] "ActiveReg"
##  [2] "NoReg"
##  [3] "SentEmails"
##  [4] "GiftAmount"
##  [5] "GiftType_offline"
##  [6] "GiftType_online"
##  [7] "PmtMethod_Cash"
##  [8] "PmtMethod_Check"
##  [9] "PmtMethod_Credit_Card"
## [10] "EmailStatus_Good"
## [11] "EmailStatus_HardBounce"
## [12] "EmailStatus_SoftBounce"
## [13] "EmailStatus_Unknown"
## [14] "Connection_Blank"
## [15] "Connection_Care_Manager_of_Person_with_MS"
## [16] "Connection_Caregiver_of_Person_with_MS"
## [17] "Connection_Child_has_MS"
## [18] "Connection_Friend_has_MS"
## [19] "Connection_FriendOrCoWorker"
## [20] "Connection_I_have_MS"
## [21] "Connection_My_friend_has_MS"
## [22] "Connection_None"
## [23] "Connection_Other"
```

```
## [24] "Connection_Parent_has_MS"
## [25] "Connection_Possible_MS"
## [26] "Connection_Relative_has_MS"
## [27] "Connection_RelativeOther"
## [28] "Connection_RelativeParent"
## [29] "Connection_Sibling_has_MS"
## [30] "Connection_Spouse_has_MS"
```

Calculate correlation matrix of variables

```
library(corrplot)

corrplot(cor(dataset), method = 'ellipse', order = 'AOE', type =
'upper', tl.cex = 0.5)
```



Prepare data for machine learning: Round One

```
set.seed(1337)
data_split <- initial_split(pre_norm_set, prop=0.7)
data_train <- data_split %>% training()
data_test <- data_split %>% testing()
```

```
dataset <- pre_norm_set %>%
  mutate(
    GiftAmount=norm(GiftAmount),
    ActiveReg=norm(ActiveReg),
    NoReg=norm(NoReg),
    SentEmails=norm(SentEmails)
  )

norm_split <- initial_split(dataset, prop=0.7)
norm_train <- norm_split %>% training()
norm_test <- norm_split %>% testing()
```

Model A1: Neural Network

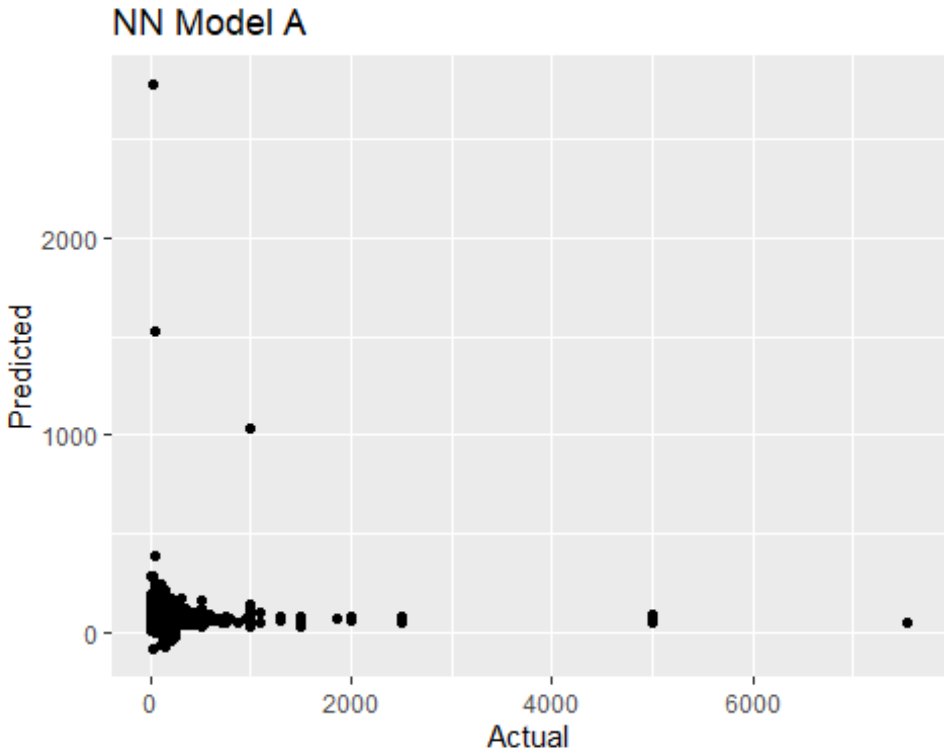
```
nnA <- neuralnet(GiftAmount ~ ., data=norm_train, hidden=c(8, 4))
plot(nnA)
```

Calculate RMSE for the first runs neural net model

```
predA <- compute(nnA, norm_test)
predA <- regular(predA$net.result, data_test$GiftAmount)
xx1 <- data_test$GiftAmount

RMSE_NN_ModelA <- (sum((xx1 - predA)^2) / length(xx1)) ^ 0.5
cat('RMSE for Neural Network Model A1: ', RMSE_NN_ModelA)

## RMSE for Neural Network Model A1:  205.8744
```

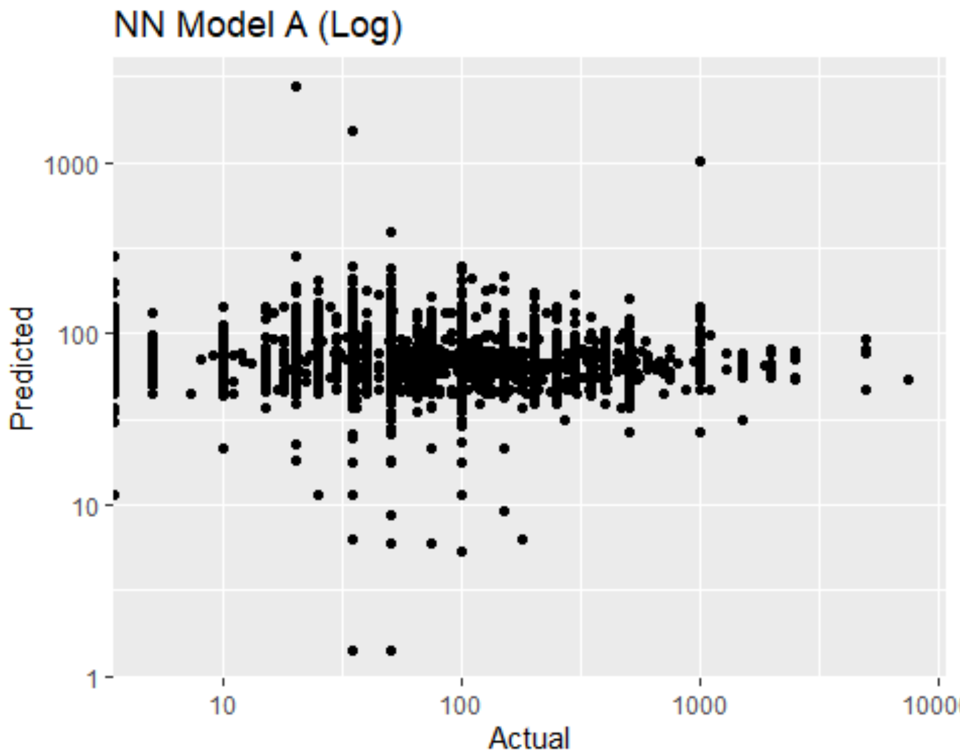
Graph the actual versus the predicted values

```
ggplot(mapping=aes(x=xx1, y=predA)) +
  geom_point() +
  labs(title="NN Model A", x="Actual", y="Predicted")
```



Create a log version of the plot above

```
options(scipen=999)
ggplot(mapping=aes(x=xx1, y=predA)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10() +
  labs(title="NN Model A (Log)", x="Actual", y="Predicted")
```



Model A2: MultiVariable Regression

```
reg_modelB1 <- lm(GiftAmount ~ ., data=data_train)
summary(reg_modelB1)

##
## Call:
## lm(formula = GiftAmount ~ ., data = data_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -288.9  -56.3  -34.8    9.8 9921.4
##
## Coefficients: (5 not defined because of singularities)
##                                                 Estimate    Std.
Error t value
## (Intercept)                                  170.69941894
13.09236756  13.038
## ActiveReg                                      0.01375394
```

```
0.00801091   1.717
## NoReg                                               -0.00004966
0.00005626  -0.883
## SentEmails                                          -0.00011639
0.00013526  -0.861
## GiftType_offline                                    57.32300928
16.10177268   3.560
## GiftType_online                                              NA
NA      NA
## PmtMethod_Cash                                    -123.48096863
25.95704983  -4.757
## PmtMethod_Check                                     16.61994292
17.47819986   0.951
## PmtMethod_Credit_Card                                        NA
NA      NA
## EmailStatus_Good                                   -31.24726219
6.27105799  -4.983
## EmailStatus_HardBounce                             -25.29824750
10.67159197  -2.371
## EmailStatus_SoftBounce                             -21.41403496
10.42808044  -2.053
## EmailStatus_Unknown                                          NA
NA      NA
## Connection_Blank                                   -54.10887240
11.46505545  -4.719
## Connection_Care_Manager_of_Person_with_MS   -9.89265472
50.56733084  -0.196
## Connection_Caregiver_of_Person_with_MS     -74.10792683
37.75256320  -1.963
## Connection_Child_has_MS                             81.77960773
16.18503705   5.053
## Connection_Friend_has_MS                           -64.53675353
11.19313232  -5.766
## Connection_FriendOrCoWorker                      -104.18684875
96.21468800  -1.083
## Connection_I_have_MS                               -41.40808981
12.79402618  -3.237
## Connection_My_friend_has_MS                                  NA
NA      NA
## Connection_None                                    -78.81194678
11.21275573  -7.029
## Connection_Other                                   -57.65289028
```

```
11.71100275  -4.923
## Connection_Parent_has_MS                         -57.18895417
14.25612143  -4.012
## Connection_Possible_MS                           -99.36380613
44.13154811  -2.252
## Connection_Relative_has_MS                       -65.64475113
11.62065798  -5.649
## Connection_RelativeOther                          49.28475545
191.46407976   0.257
## Connection_RelativeParent                        282.80691725
191.58102352   1.476
## Connection_Sibling_has_MS                        -41.39739311
13.78697895  -3.003
## Connection_Spouse_has_MS                                   NA
NA      NA
##                                                             Pr(>|t|)
## (Intercept)                                      < 0.0000000000000002 ***
## ActiveReg                                                    0.086010 .
## NoReg                                                        0.377342
## SentEmails                                                   0.389506
## GiftType_offline                                             0.000371 ***
## GiftType_online                                                    NA
## PmtMethod_Cash                                       0.00000197484049 ***
## PmtMethod_Check                                              0.341667
## PmtMethod_Credit_Card                                              NA
## EmailStatus_Good                                     0.00000063105206 ***
## EmailStatus_HardBounce                                       0.017766 *
## EmailStatus_SoftBounce                                       0.040035 *
## EmailStatus_Unknown                                                NA
## Connection_Blank                                     0.00000237774627 ***
## Connection_Care_Manager_of_Person_with_MS                    0.844899
## Connection_Caregiver_of_Person_with_MS                       0.049659 *
## Connection_Child_has_MS                              0.00000043852337 ***
## Connection_Friend_has_MS                             0.00000000822721 ***
## Connection_FriendOrCoWorker                                  0.278882
## Connection_I_have_MS                                         0.001212 **
## Connection_My_friend_has_MS                                        NA
## Connection_None                                      0.00000000000214 ***
## Connection_Other                                     0.00000085794296 ***
## Connection_Parent_has_MS                             0.00006050348137 ***
```

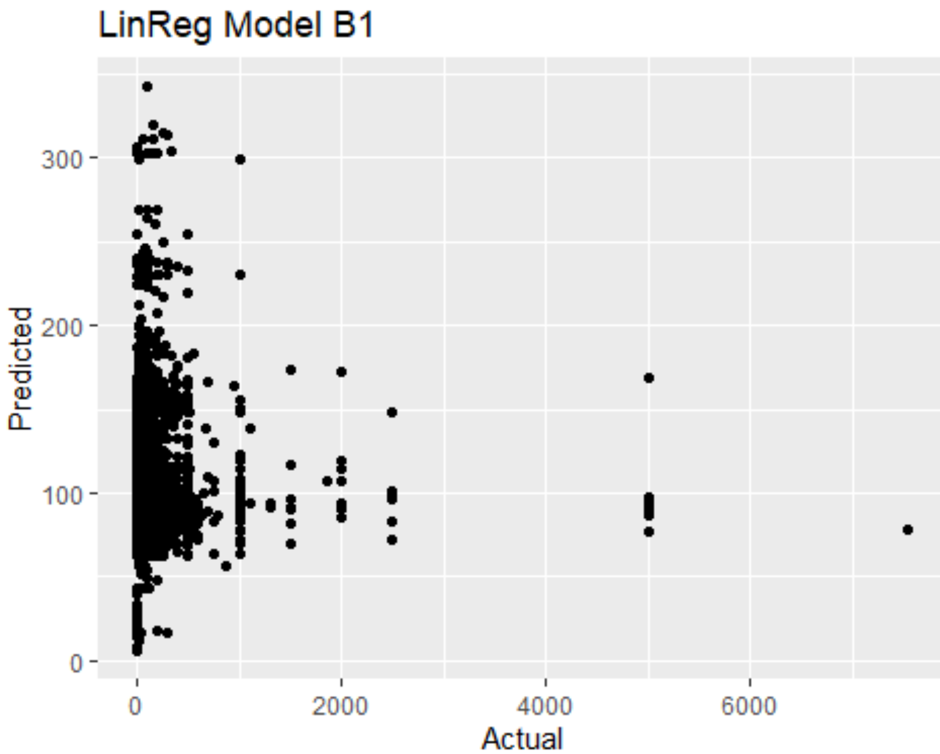
```
## Connection_Possible_MS                                          0.024360 *
## Connection_Relative_has_MS                              0.00000001631979 ***
## Connection_RelativeOther                                        0.796864
## Connection_RelativeParent                                       0.139910
## Connection_Sibling_has_MS                                       0.002679 **
## Connection_Spouse_has_MS                                              NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 191.1 on 24544 degrees of freedom
## Multiple R-squared:  0.01865,    Adjusted R-squared:  0.01769
## F-statistic: 19.44 on 24 and 24544 DF,  p-value: <
0.00000000000000022
```

Compute the RMSE of the Regression model

```
predB1 <- reg_modelB1 %>% predict(data_test)

## Warning in predict.lm(., data_test): prediction from a rank-
deficient fit may be
## misleading

rt <- data_test$GiftAmount


RMSE_RegModelB1 <- (sum((rt - predB1)^2)/length(rt))^0.5
cat('RMSE for Regression Model B1: ', RMSE_RegModelB1)

## RMSE for Regression Model B1:  201.0432
```

Graph the results of the first regression model

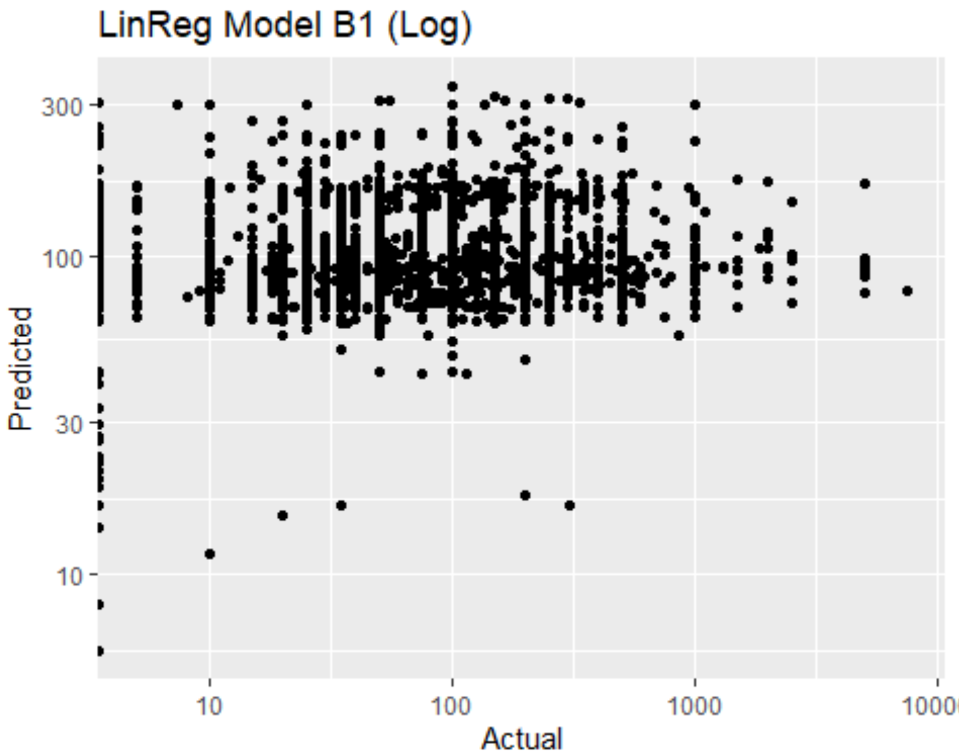
```
ggplot(mapping=aes(x=rt, y=predB1)) +
  geom_point() +
  labs(title="LinReg Model B1", x="Actual", y="Predicted")
```



Graph the previous graph using the log scale

```
ggplot(mapping=aes(x=rt, y=predB1)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10() +
  labs(title="LinReg Model B1 (Log)", x="Actual", y="Predicted")
```

```
## Warning: Transformation introduced infinite values in continuous x-
axis
```



Print all significant variables

```
coef1 <-
data.frame(summary(reg_modelB1)$coef[summary(reg_modelB1)$coef[,4] <=
.05, 4])
coef1
```

```
##
summary.reg_modelB1..coef.summary.reg_modelB1..coef...4.....0.05..
## (Intercept)
0.00000000000000000000000000000000000009993509
## GiftType_offline
0.00037149408912503151030878867366880058398237 5
## PmtMethod_Cash
0.00000197484048728124473052678022888528630574 0
## EmailStatus_Good
0.00000631052060167988270117725435603972528043
## EmailStatus_HardBounce
0.017766146402847945995340239733195630833506584
## EmailStatus_SoftBounce
```

```
0.040034944352191369210114402221734053455293179
## Connection_Blank
0.000002377746271808022360825032870401685158868
## Connection_Caregiver_of_Person_with_MS
0.049658541961133541298156757193282828666269779
## Connection_Child_has_MS
0.00000438523368858389779936790553449554863619
## Connection_Friend_has_MS
0.000000008227208794569809290411765978490166162
## Connection_I_have_MS
0.001211582393394273430425878146365903376135975
## Connection_None
0.000000000002138039656640298436180228081049393
## Connection_Other
0.000000857942959897780972611247563008873839863
## Connection_Parent_has_MS
0.000060503481370606806925144249831305387488101
## Connection_Possible_MS
0.024360362539931124103986803675070405006408691
## Connection_Relative_has_MS
0.000000016319790041661453601669340418567344386
## Connection_Sibling_has_MS
0.002679145406837397502125552861684809613507241
```

Machine Learning Round Two

Run your second neural network model

```
nnB <- neuralnet(GiftAmount ~ GiftType_offline + PmtMethod_Cash +
EmailStatus_Good + EmailStatus_HardBounce + EmailStatus_SoftBounce +
Connection_Blank + Connection_Caregiver_of_Person_with_MS +
Connection_Child_has_MS + Connection_Friend_has_MS +
Connection_I_have_MS + Connection_None + Connection_Other +
Connection_Parent_has_MS + Connection_Possible_MS +
Connection_Relative_has_MS + Connection_Sibling_has_MS,
data=norm_train, hidden=c(8, 4))
plot(nnB)
```

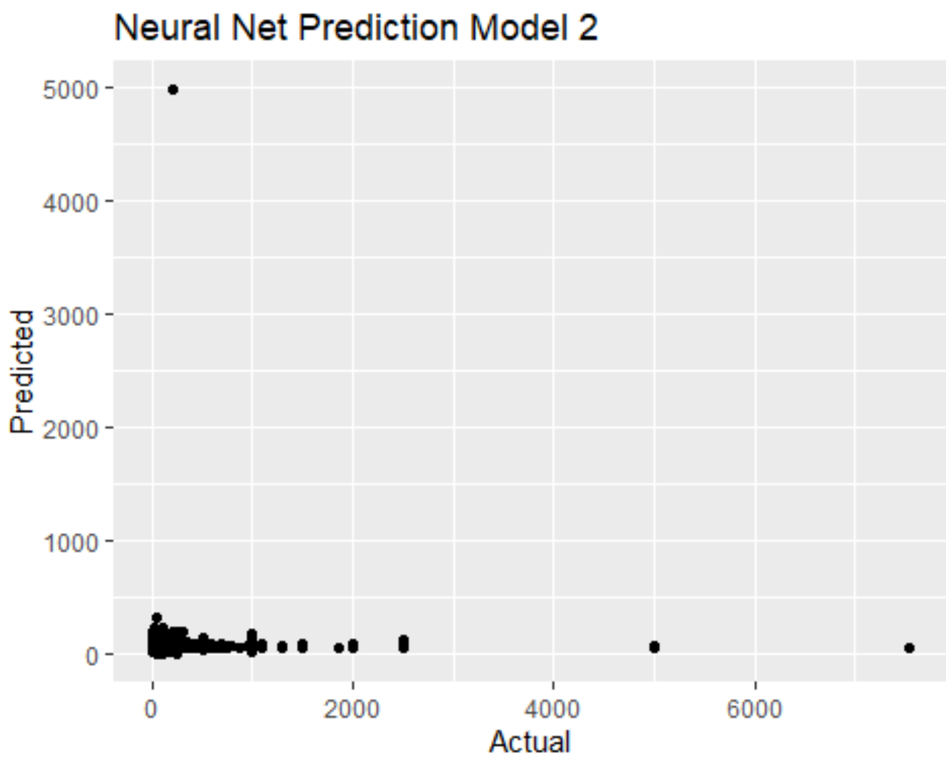
Calculate the RMSE for the second neural network

```
predA2 <- compute(nnB, norm_test)
predA2 <- regular(predA2$net.result, data_test$GiftAmount)
xx12 <- data_test$GiftAmount
```

```
RMSE_NN_ModelA2 <- (sum((xx12 - predA2)^2) / length(xx12)) ^ 0.5
cat('RMSE for Neural Network Model A2: ', RMSE_NN_ModelA2)

## RMSE for Neural Network Model A2:  208.6993
```

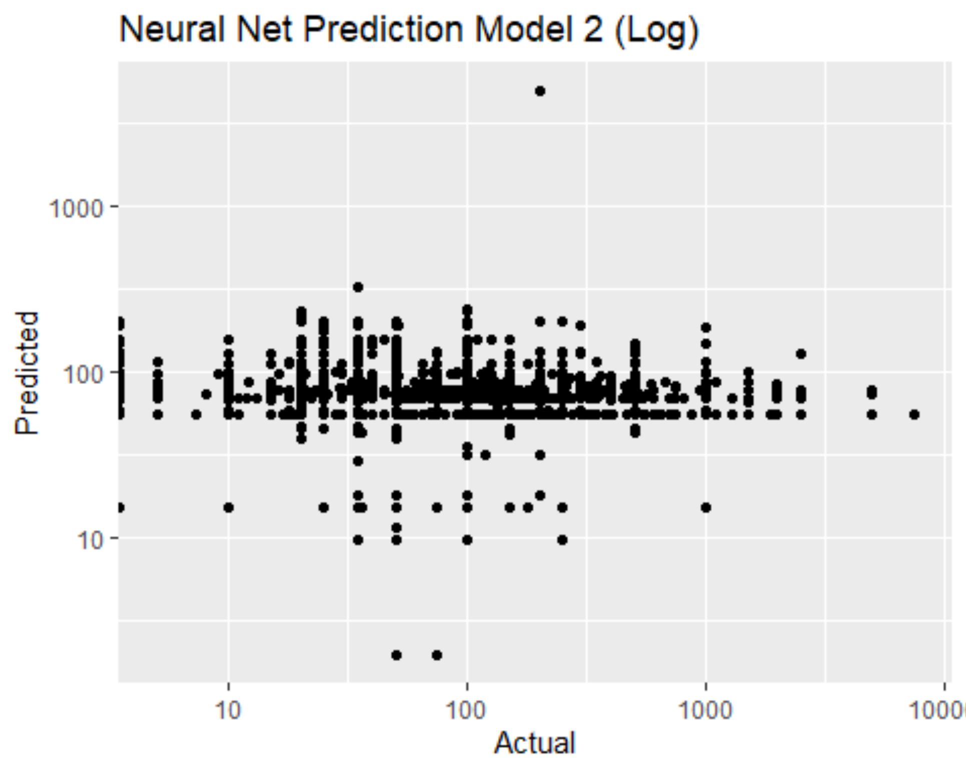
Graph the actual and expected variables based off the second neural network model

```
ggplot(mapping=aes(x=xx12, y=predA2)) +
  geom_point() +
  labs(title="Neural Net Prediction Model 2", x="Actual",
y="Predicted")
```



Log plot of neural network model 2

```
ggplot(mapping=aes(x=xx12, y=predA2)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10() +
  labs(title="Neural Net Prediction Model 2 (Log)", x="Actual",
y="Predicted")
```



Calculate your second linear regression with the significant variables

```
reg_modelB2 <- lm(GiftAmount ~ GiftType_offline + PmtMethod_Cash +
EmailStatus_Good + EmailStatus_HardBounce + EmailStatus_SoftBounce +
Connection_Blank + Connection_Caregiver_of_Person_with_MS +
Connection_Child_has_MS + Connection_Friend_has_MS +
Connection_I_have_MS + Connection_None + Connection_Other +
Connection_Parent_has_MS + Connection_Possible_MS +
Connection_Relative_has_MS + Connection_Sibling_has_MS,
data=data_train)
summary(reg_modelB2)

##
## Call:
## lm(formula = GiftAmount ~ GiftType_offline + PmtMethod_Cash +
##     EmailStatus_Good + EmailStatus_HardBounce +
EmailStatus_SoftBounce +
##     Connection_Blank + Connection_Caregiver_of_Person_with_MS +
```

```
##     Connection_Child_has_MS + Connection_Friend_has_MS +
Connection_I_have_MS +
##     Connection_None + Connection_Other + Connection_Parent_has_MS +
##     Connection_Possible_MS + Connection_Relative_has_MS +
Connection_Sibling_has_MS,
##     data = data_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -294.5  -54.5  -37.1   12.9 9912.9
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                               182.909     12.180  15.017
## GiftType_offline                           70.500      6.492  10.860
## PmtMethod_Cash                           -139.227     21.357  -6.519
## EmailStatus_Good                          -31.508      6.267  -5.028
## EmailStatus_HardBounce                    -25.707     10.666  -2.410
## EmailStatus_SoftBounce                    -21.630     10.419  -2.076
## Connection_Blank                          -53.592     11.114  -4.822
## Connection_Caregiver_of_Person_with_MS  -74.097     37.658  -1.968
## Connection_Child_has_MS                    82.589     15.944   5.180
## Connection_Friend_has_MS                  -64.263     10.841  -5.928
## Connection_I_have_MS                      -41.002     12.484  -3.284
## Connection_None                           -78.887     10.862  -7.263
## Connection_Other                          -57.684     11.376  -5.071
## Connection_Parent_has_MS                  -56.874     13.981  -4.068
## Connection_Possible_MS                   -100.244     44.051  -2.276
## Connection_Relative_has_MS                -65.239     11.282  -5.783
## Connection_Sibling_has_MS                 -41.354     13.502  -3.063
##                                                       Pr(>|t|)
## (Intercept)                              < 0.0000000000000002 ***
## GiftType_offline                         < 0.0000000000000002 ***
## PmtMethod_Cash                               0.00000000007211 ***
## EmailStatus_Good                             0.00000049924061 ***
## EmailStatus_HardBounce                                0.01595 *
## EmailStatus_SoftBounce                                0.03791 *
## Connection_Blank                             0.00000143065624 ***
## Connection_Caregiver_of_Person_with_MS                0.04912 *
## Connection_Child_has_MS                      0.00000022358099 ***
```

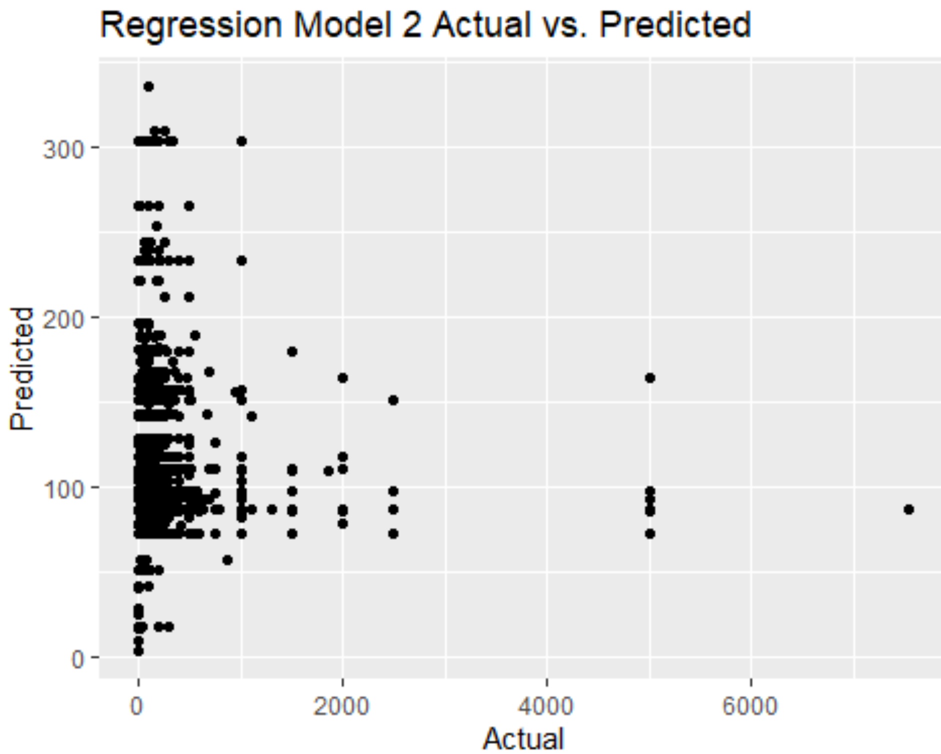
```
## Connection_Friend_has_MS                        0.00000000310749 ***
## Connection_I_have_MS                                     0.00102 **
## Connection_None                                 0.00000000000039 ***
## Connection_Other                                0.00000039891191 ***
## Connection_Parent_has_MS                        0.00004760418717 ***
## Connection_Possible_MS                                   0.02288 *
## Connection_Relative_has_MS                      0.00000000744761 ***
## Connection_Sibling_has_MS                                0.00220 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 191.2 on 24552 degrees of freedom
## Multiple R-squared:  0.0179, Adjusted R-squared:  0.01726
## F-statistic: 27.97 on 16 and 24552 DF,  p-value: <
0.00000000000000022
```

Calculate RMSE off you second regression model

```
predB2 <- reg_modelB2 %>% predict(data_test)

rt2 <- data_test$GiftAmount

RMSE_RegModelB2 <- (sum((rt2 - predB2)^2)/length(rt2))^0.5
cat('RMSE for Regression Model B2: ', RMSE_RegModelB2)

## RMSE for Regression Model B2:  201.0957
```

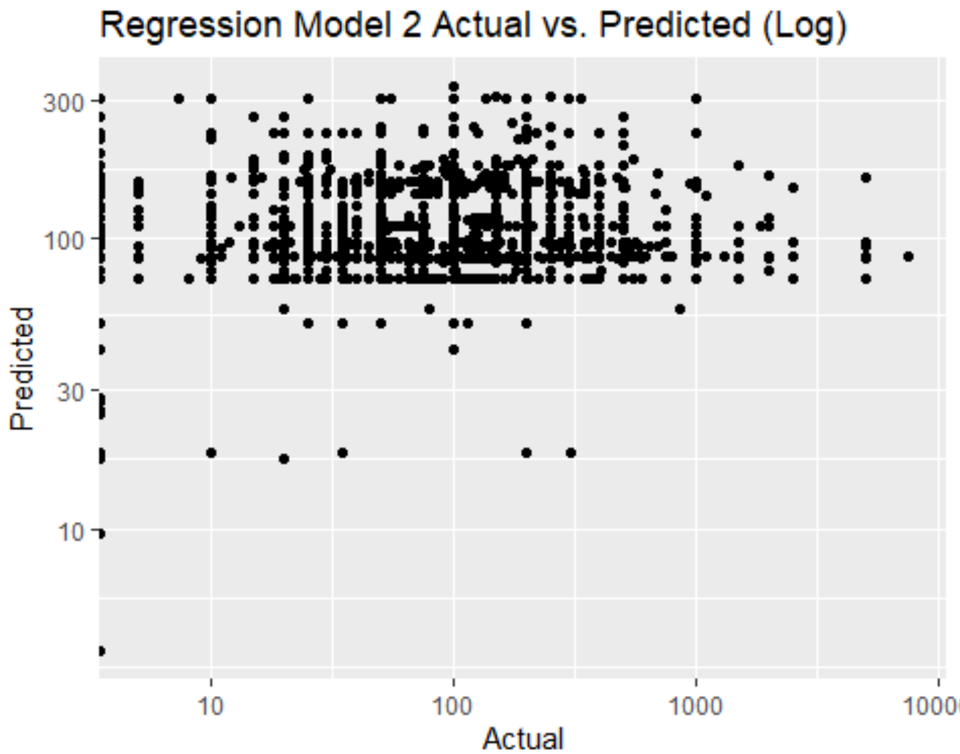
Plot the actual vs. predicted values for your second regression

```
ggplot(mapping=aes(x=rt2, y=predB2)) +
  geom_point() +
  labs(title="Regression Model 2 Actual vs. Predicted", x="Actual",
y="Predicted")
```



Give the log plot of the plot above

```
ggplot(mapping=aes(x=rt2, y=predB2)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10() +
  labs(title="Regression Model 2 Actual vs. Predicted (Log)",
x="Actual", y="Predicted")

## Warning: Transformation introduced infinite values in continuous x-
axis
```



Ensemble Method: Combining Neural Network With Decison Tree

```
ensemble_set <- pre_norm_set %>%
  select(ActiveReg, NoReg, SentEmails,GiftAmount, EmailStatus_Good,
Connection_Friend_has_MS)

set.seed(567)
ensemble_split <- initial_split(ensemble_set, prop=0.7)
ensemble_train <- ensemble_split %>% training()
ensemble_test <- ensemble_split %>% testing()

ensemble_set <- ensemble_set %>%
  mutate(
    ActiveReg=norm(ActiveReg),
    NoReg=norm(NoReg),
    SentEmails=norm(SentEmails),
    GiftAmount=norm(GiftAmount)
  )
```

```r
norm_e_split <- initial_split(ensemble_set, prop=0.7)
norm_e_train <- norm_e_split %>% training()
norm_e_test <- norm_e_split %>% testing()
```

Build a neural network model

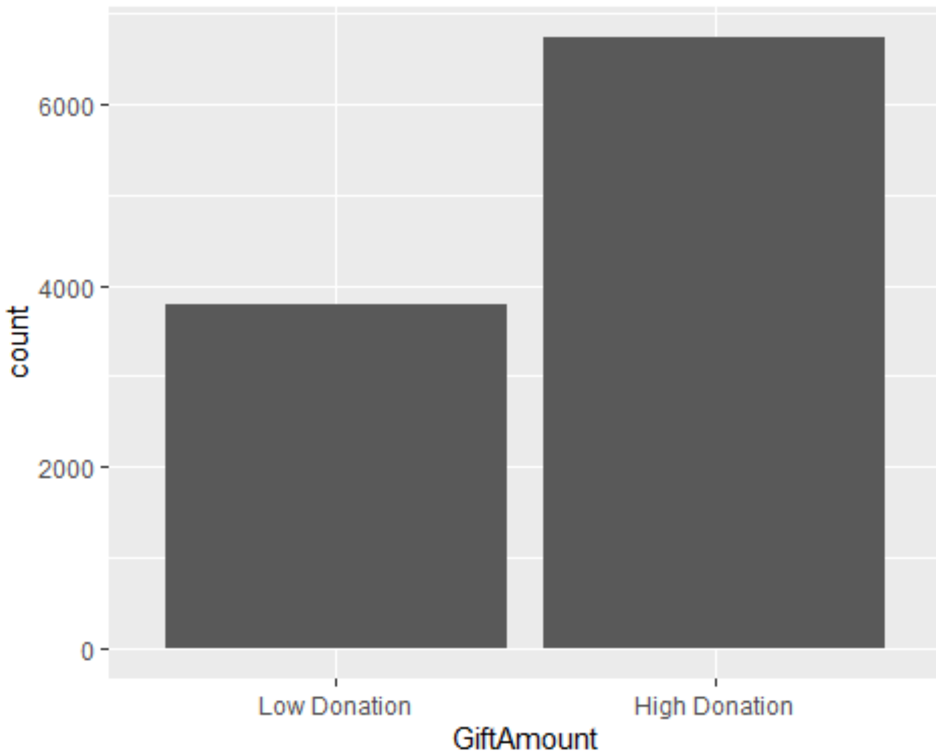
```r
nnC <- neuralnet(GiftAmount ~ ., data=norm_e_train, hidden=c(5, 3))
plot(nnC)
```

Alter the dataset to hold the predictions

```r
predictions <- compute(nnC, norm_e_test)
netResults <- regular(predictions$net.result,
ensemble_test$GiftAmount)

# Replace the actual amount with the predicted amount, bin Gift Amount

dset <- ensemble_test %>%
  select(-GiftAmount) %>%
  mutate(GiftAmount=cut(netResults, breaks=c(-1, 90, 150),
labels=c("Low Donation", "High Donation")))

table(unlist(dset[, c("GiftAmount")]))

##
##  Low Donation High Donation
##          3789          6741
```

Plot Histogram of Gift Amount Class

```r
dset %>%
  ggplot(mapping=aes(x=GiftAmount)) +
  geom_histogram(stat="count")
```

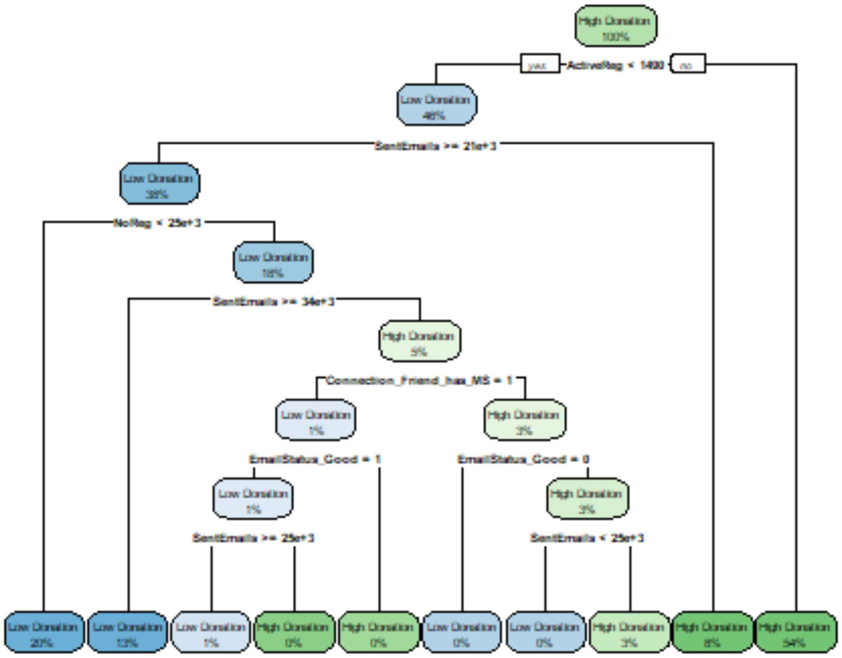


Split new dataset

```
#set.seed(999)
tree_split <- initial_split(dset, prop=0.7)
tree_train <- tree_split %>% training()
tree_test <- tree_split %>% testing()
```

Train decision tree

```
library(rpart)

library(rpart.plot, warn.conflicts=FALSE)

fit <- rpart(GiftAmount ~ ., data=tree_train,
method="class",control=rpart.control(cp=0))

rpart.plot(fit, extra=100)
```



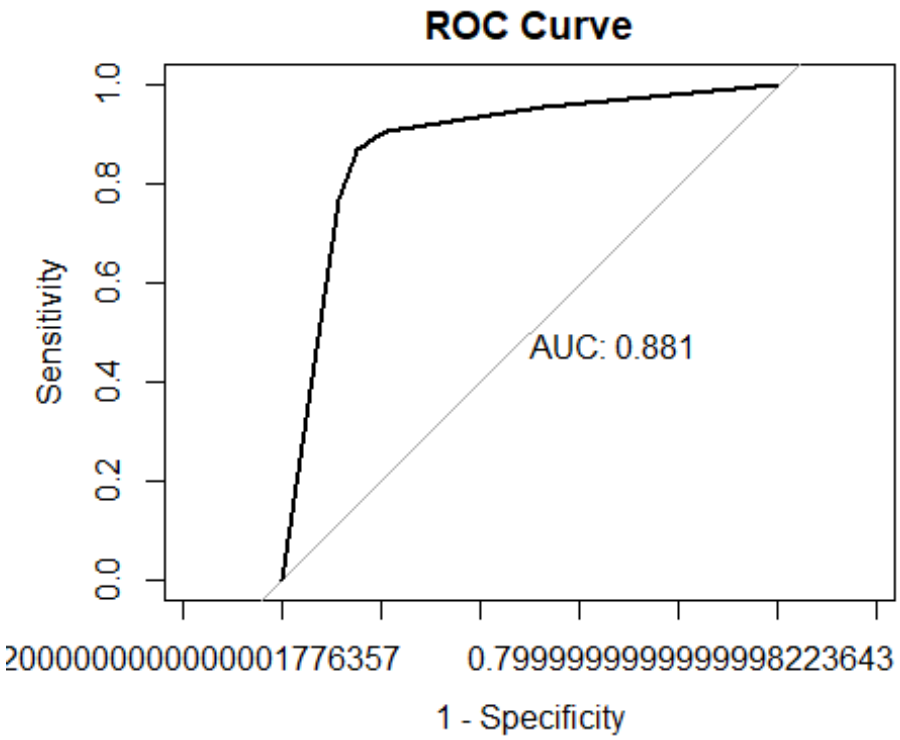
Generate Classifications from the Decision Tree (Confusion Matrix)

```
prediction <- predict(fit, tree_test, type='class')
confusionMatrix(prediction, tree_test$GiftAmount, mode="everything")

## Confusion Matrix and Statistics
##
##                  Reference
## Prediction        Low Donation High Donation
##   Low Donation             928           219
##   High Donation            213          1800
##
##                  Accuracy : 0.8633
##                    95% CI : (0.8508, 0.8751)
##     No Information Rate : 0.6389
##     P-Value [Acc > NIR] : <0.0000000000000002
##
##                     Kappa : 0.704
##
```

```
##  Mcnemar's Test P-Value : 0.8099
##
##             Sensitivity : 0.8133
##             Specificity : 0.8915
##          Pos Pred Value : 0.8091
##          Neg Pred Value : 0.8942
##               Precision : 0.8091
##                  Recall : 0.8133
##                      F1 : 0.8112
##              Prevalence : 0.3611
##          Detection Rate : 0.2937
##    Detection Prevalence : 0.3630
##       Balanced Accuracy : 0.8524
##
##        'Positive' Class : Low Donation
##
```

Compute ROC Curve and AUC

```
library(pROC)

prob_pred <- predict(fit, tree_test, type='prob')[,2]

roc_curve <- roc(tree_test$GiftAmount, prob_pred)

## Setting levels: control = Low Donation, case = High Donation

## Setting direction: controls < cases

plot(roc_curve, main="ROC Curve", print.auc=TRUE, legacy.axes=TRUE,
revC=TRUE)
```




```
auc(roc_curve)

## Area under the curve: 0.8813
```