Project 1 Annotated Code

Gavin Gunawardena

Original Dataset Configuration and Analysis

```
#Load and Install(If not present), all packages used in this project
# function to install packages if they don't exist
usePackage <- function(p) {
    if (!is.element(p, installed.packages()[,1]))
        install.packages(p, dep = TRUE)
    require(p, character.only = TRUE)
}
# load packages used in this project
usePackage("readxl")

## Loading required package: readxl

usePackage("ggplot2")

## Loading required package: ggplot2

usePackage("caret")

## Loading required package: caret

## Loading required package: lattice

# Read excel files into dataframes. The below code assumes that the excel
files are at the same file location as this R file.
Subterraneus.df <- as.data.frame(read_excel("Vole Skulls.xlsm",
sheet="Subterraneus"))
Multiplex.df <- as.data.frame(read_excel("Vole Skulls.xlsm",
sheet="Multiplex"))
Unknown.df <- as.data.frame(read_excel("Vole Skulls.xlsm", sheet="Unknown"))

#Standardize column names for easier use.
OrigColNames <- colnames(Subterraneus.df)
StandColNames <- c("V_S_ID", "ChromosomalID", "IncisiveOrSkullLength",
"SkullHeight", "SkullWidth")
colnames(Subterraneus.df) <- StandColNames
colnames(Multiplex.df) <- StandColNames
colnames(Unknown.df) <- StandColNames

#Remove rows with no values.
Subterraneus.df <- Subterraneus.df[rowSums(is.na(Subterraneus.df)) == 0, ]
Multiplex.df <- Multiplex.df[rowSums(is.na(Multiplex.df)) == 0, ]
Unknown.df <- Unknown.df[rowSums(is.na(Unknown.df)) == 0, ]
```

```r
# Combine the Subterraneus.df and Multiplex.df datasets for use in model
training
Combined_S_M.df <- merge(Subterraneus.df,Multiplex.df, all = TRUE)


# Update the dataframe with subterraneus and multiplex samples to have an
extra column representing ChromosomalID by 1(subterraneus) and 0(multiplex)
to improve the speed of logistic regression model training and unlock the
logit parameter for the binary option in the glm function.
Combined_S_M.df$ChromosomalIDBinary <-
as.factor(ifelse(Combined_S_M.df$ChromosomalID == "subterraneus", 1, 0))

Combined_S_M_U.df <- merge(Combined_S_M.df,Unknown.df, all = TRUE)

#create par plots, summary tables, and box plots of the 3 main datasets,
Subterraneus.df, Multiplex.df, Unknown.df.

par(mfrow = c(2,3))

# par plots
plot(Subterraneus.df[,2:5])
title(main="Subterraneus Dataset")

print("Subterraneus Data Summary")
summary(Subterraneus.df[,2:5])


plot(Multiplex.df[,2:5])
title(main="Multiplex Dataset")

print("Multiplex Data Summary")
summary(Multiplex.df[,2:5])

plot(Unknown.df[,2:5])
title(main="Unknown Dataset")

print("Unknown Data Summary")
summary(Unknown.df[,2:5])


# box plot to look for outliers in the known data
ggplot(Combined_S_M_U.df, aes(x=ChromosomalID, y=IncisiveOrSkullLength)) +
  geom_boxplot() + ylab(label = "Incisive or Skull Length")

ggplot(Combined_S_M_U.df, aes(x=ChromosomalID, y=SkullHeight)) +
  geom_boxplot() + ylab(label = "Skull Height")

ggplot(Combined_S_M_U.df, aes(x=ChromosomalID, y=SkullWidth)) +
  geom_boxplot() + ylab(label = "Skull Width")
```

Look for and remove the outliers in the data.

```
#After some analysis, I've noticed that the outliers are all around 10 times
larger or smaller than the rest of the samples, and the additional attributes
of each individual sample that has an outlier, are not proportionally larger
or smaller based on the outlier, but are close to the mean of the data. This
has led me to believe that the outliers are data entry or measurement errors,
worthy of being removed from the dataset. I decided not to remove the
outliers from the unclassified data, since the main purpose of removing the
outliers is to improve the logistical model which I'll be using to classify
the unclassified data.


# Find the outliers based on whether a datapoint is 5 times higher or lower
than the mean of the feature in the dataset
subset(Subterraneus.df, IncisiveOrSkullLength >=
mean(IncisiveOrSkullLength)*5 | IncisiveOrSkullLength <=
mean(IncisiveOrSkullLength)/5)
subset(Subterraneus.df, SkullHeight >= mean(SkullHeight)*5 | SkullHeight <=
mean(SkullHeight)/5)
subset(Subterraneus.df, SkullWidth >= mean(SkullWidth)*5 | SkullWidth <=
mean(SkullWidth)/5)

subset(Multiplex.df, IncisiveOrSkullLength >= mean(IncisiveOrSkullLength)*5 |
IncisiveOrSkullLength <= mean(IncisiveOrSkullLength)/5)
subset(Multiplex.df, SkullHeight >= mean(SkullHeight)*5 | SkullHeight <=
mean(SkullHeight)/5)
subset(Multiplex.df, SkullWidth >= mean(SkullWidth)*5 | SkullWidth <=
mean(SkullWidth)/5)

subset(Unknown.df, IncisiveOrSkullLength >= mean(IncisiveOrSkullLength)*5 |
IncisiveOrSkullLength <= mean(IncisiveOrSkullLength)/5)
subset(Unknown.df, SkullHeight >= mean(SkullHeight)*5 | SkullHeight <=
mean(SkullHeight)/5)
subset(Unknown.df, SkullWidth >= mean(SkullWidth)*5 | SkullWidth <=
mean(SkullWidth)/5)



# Create a function to quickly remove the outliers
Remove_Outliers_Vole_Datasets <- function(x)
{
  x <- subset(x, IncisiveOrSkullLength <= mean(IncisiveOrSkullLength)*5 &
IncisiveOrSkullLength >= mean(IncisiveOrSkullLength)/5)
  x <- subset(x, SkullHeight <= mean(SkullHeight)*5 & SkullHeight >=
mean(SkullHeight)/5)
  x <- subset(x, SkullWidth <= mean(SkullWidth)*5 & SkullWidth >=
mean(SkullWidth)/5)
}

# Use the function for removing the outliers
Subterraneus.df <- Remove_Outliers_Vole_Datasets(Subterraneus.df)
Multiplex.df <- Remove_Outliers_Vole_Datasets(Multiplex.df)



# Update the combined datasets

Combined_S_M.df <- merge(Subterraneus.df,Multiplex.df, all = TRUE)



# Update the dataframe with subterraneus and multiplex samples to have an
extra column representing ChromosomalID by 1(subterraneus) and 0(multiplex).
The purpose of this is to speed up the training and testing of the model by
classifying the data with numbers instead of strings.
Combined_S_M.df$ChromosomalIDBinary <-
as.factor(ifelse(Combined_S_M.df$ChromosomalID == "subterraneus", 1, 0))
```

Model selection and analysis

```
#Create models and adjust them based on their summaries
model1 <- glm(ChromosomalIDBinary ~
IncisiveOrSkullLength*SkullHeight*SkullWidth, data = Combined_S_M.df, family
= binomial(logit))
model2 <- glm(ChromosomalIDBinary ~
I(IncisiveOrSkullLength^2)*I(SkullHeight^2)*I(SkullWidth^2), data =
Combined_S_M.df, family = binomial(logit))
model3 <- glm(ChromosomalIDBinary ~
IncisiveOrSkullLength*SkullHeight*SkullWidth+1, data = Combined_S_M.df,
family = binomial(logit))
model4 <- glm(ChromosomalIDBinary ~
log(IncisiveOrSkullLength)*SkullHeight*SkullWidth, data = Combined_S_M.df,
family = binomial(logit))



#Summarize the models to compare them.
summary(model1)

summary(model2)

summary(model3)

summary(model4)
```

Use of the Model with Repeated 10 times 10 fold – Leave One Out Cross Validation

```
#specify the cross-validation method
ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
```

```
#Fit a regression model and use LOOCV to evaluate performance
model1.5_fit <- train(form=ChromosomalIDBinary ~ IncisiveOrSkullLength +
SkullHeight + SkullWidth, data = Combined_S_M.df, method = "glm", trControl =
ctrl)

#view summary results of the model

print(model1.5_fit$results)



#Plot what the model should look like to scatter plots of the data and with a
line showing the logistic regression curve:

par(mfrow=c(1,3))
#plot logistic regression curve
ggplot(Combined_S_M.df,
aes(x=IncisiveOrSkullLength,as.numeric(ChromosomalIDBinary) - 1)) +
  geom_point(alpha=.5) +
    stat_smooth(method="glm", se=FALSE, method.args = list(family=binomial))
+
    labs(x = "Incisive Or Skull Length", y = "Multiplex(0) or
Subterraneus(1)", Title = "logistic regression curve")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
#plot logistic regression curve
ggplot(Combined_S_M.df, aes(x=SkullHeight,as.numeric(ChromosomalIDBinary) -
1)) +
  geom_point(alpha=.5) +
    stat_smooth(method="glm", se=FALSE, method.args = list(family=binomial))
+
    labs(x = "Skull Height", y = "Multiplex(0) or Subterraneus(1)", Title =
"logistic regression curve")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
#plot logistic regression curve
  ggplot(Combined_S_M.df, aes(SkullWidth,as.numeric(ChromosomalIDBinary) -
1)) +
  geom_point(alpha=.5) +
    stat_smooth(method="glm", se=FALSE, method.args = list(family=binomial))
+
    labs(x = "Skull Width", y = "Multiplex(0) or Subterraneus(1)", Title =
"logistic regression curve")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
#Create a new dataframe to output the classified results to
Unknown.classified.df <- Unknown.df
```

```r
#Run the model on the unknown vole skull data, outputing the results to the
newly created dataframe for classified results
Unknown.classified.df$ChromosomalIDBinary <- predict(model1.5_fit,newdata
=Unknown.df,type="raw")


#Update the Unknown.classified.df so that it mimics the original dataset
column names and columns  to get it ready for output into a csv file
#Convert the binary model output into the Subterraneus and Multiplex
categories
Unknown.classified.df$ChromosomalID <-
ifelse(Unknown.classified.df$ChromosomalIDBinary == 1, "Subterraneus",
"Multiplex")

#Remove unnecessary columns
Unknown.classified.df <- Unknown.classified.df[,1:5]

#Give the columns the names they originally had in the original Excel file
colnames(Unknown.classified.df) <- OrigColNames

# Output the results to a csv file
write.csv(Unknown.classified.df[,1:5],"Vole Skulls Unknown Classified.csv",
row.names = FALSE)
```