MVA_FINAL_SAURABH_SANKHE

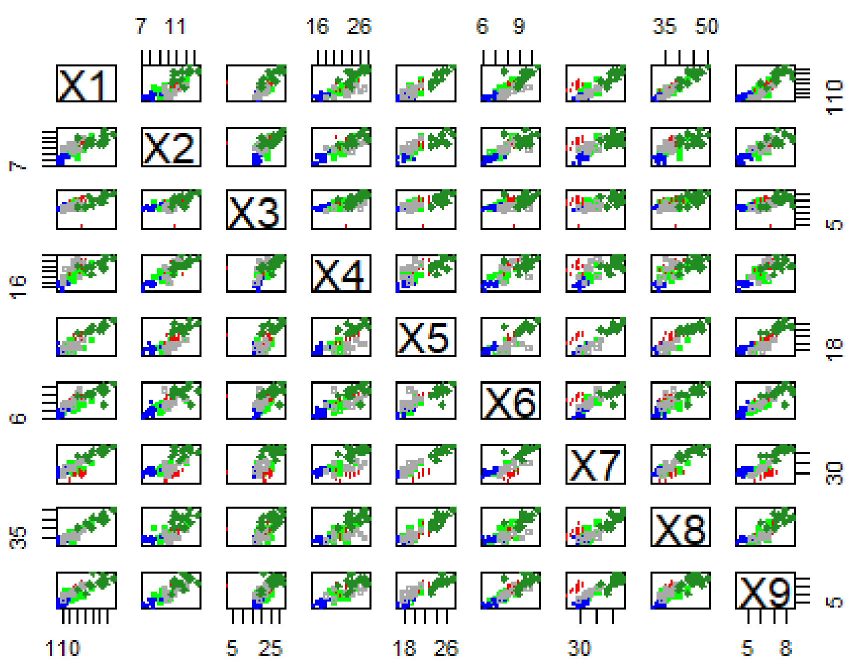
Saurabh Sankhe

April 26, 2019

Answer1:

```
#Loading the dataset
df <- read.delim("C:/Users/Saurabh/Desktop/Sem-2 Course
Documents/Multivariate Analysis/Lectures/Final/Dog_Data.txt")

#Printing the dummary of data
summary(df)

##        CanineGroup        X1              X2                X3
##  Cuons        :17   Min.   :105   Min.   : 7.200   Min.   : 2.00
##  GoldenJackal:20   1st Qu.:114   1st Qu.: 8.700   1st Qu.:19.00
##  IndianWolves:14   Median :125   Median :10.000   Median :21.00
##  ModernDog    :16   Mean   :129   Mean   : 9.961   Mean   :21.64
##  ThaiDogs     :10   3rd Qu.:137   3rd Qu.:10.900   3rd Qu.:25.00
##                     Max.   :177   Max.   :13.400   Max.   :32.00
##        X4               X5               X6              X7
##  Min.   :15.00   Min.   :17.00   Min.   : 6.0   Min.   :26.00
##  1st Qu.:18.00   1st Qu.:19.00   1st Qu.: 7.1   1st Qu.:30.00
##  Median :22.00   Median :20.00   Median : 7.9   Median :31.00
##  Mean   :21.49   Mean   :20.49   Mean   : 8.0   Mean   :32.52
##  3rd Qu.:24.00   3rd Qu.:22.00   3rd Qu.: 8.7   3rd Qu.:33.00
##  Max.   :28.00   Max.   :27.00   Max.   :10.5   Max.   :43.00
##        X8              X9              Gender
##  Min.   :31.0   Min.   :4.300   Female :32
##  1st Qu.:34.0   1st Qu.:5.300   Male   :35
##  Median :36.0   Median :6.100   Unknown:10
##  Mean   :37.4   Mean   :6.075
##  3rd Qu.:39.0   3rd Qu.:6.800
##  Max.   :50.0   Max.   :8.500

#Converting CanioneGroup to factor variable
df$CanineGroup <- as.factor(df$CanineGroup)

#Printig the levels of Canine Group
levels(df$CanineGroup)

## [1] "Cuons"        "GoldenJackal" "IndianWolves" "ModernDog"
## [5] "ThaiDogs"

#Changing Gender to Factor variable
df$Gender <- as.factor(df$Gender)
```

```r
#Printing the Levels of Gender to Console
levels(df$Gender)

## [1] "Female"  "Male"    "Unknown"

#Checking different groups for Canine
levels(df$CanineGroup)

## [1] "Cuons"        "GoldenJackal" "IndianWolves" "ModernDog"
## [5] "ThaiDogs"

#Assigning different colors as per CanineGroup
my_cols <- c("#FF0000", "#0000FF","#228B22","#00FF00","#A9A9A9")


#Printing Scatterplot for X1 to X9
pairs(df[,c(2:10)], pch = c(1,16,9,12,14)[as.numeric(df$CanineGroup)], cex =
0.5, col = my_cols[df$CanineGroup])
```



Answer2:

```r
#Calculating Distance Matrix
dist.df <- dist(df[,c(2:10)],method='euclidean')
```

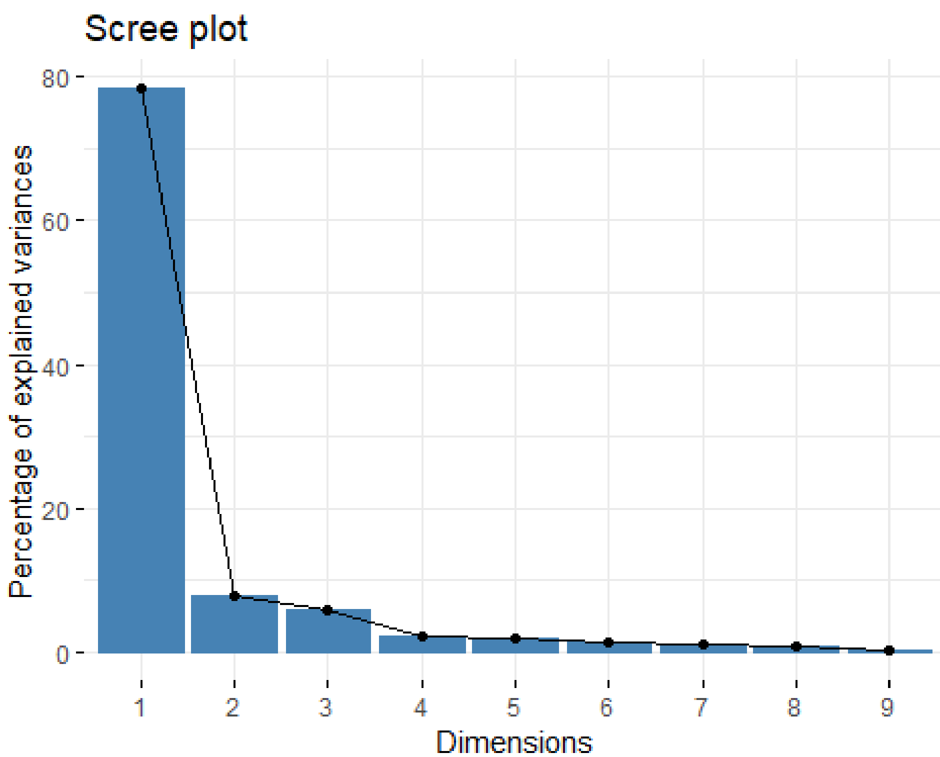
Answer3:

```
#Applying PCA function on the dataset
pca.df <- prcomp(df[,c(2:10)], scale=TRUE)

#Printing the results of pca to console
pca.df

## Standard deviations (1, .., p=9):
## [1] 2.6555963 0.8391652 0.7365758 0.4390554 0.4241988 0.3627806 0.3031519
## [8] 0.2652189 0.1857418
##
## Rotation (n x k) = (9 x 9):
##          PC1         PC2         PC3         PC4         PC5         PC6
## X1 0.3636408 -0.11451510  0.08210471 -0.30326354  0.24950692 -0.07899550
## X2 0.3424554  0.31490128 -0.19979188  0.33605928  0.01517931  0.49451257
## X3 0.2665621  0.32018675  0.87894338  0.04161625 -0.18169514 -0.04568559
## X4 0.3265349  0.44638084 -0.16540131  0.26534253  0.54545187 -0.21526217
## X5 0.3539586 -0.14160855 -0.03861441 -0.26352534 -0.33012092  0.43239890
## X6 0.3459444  0.06792334 -0.26250857  0.05378069 -0.51974026 -0.68294862
## X7 0.2859405 -0.68736531  0.13651981  0.64014932  0.05443187 -0.01170970
## X8 0.3470802 -0.28877388  0.03666665 -0.47256682  0.40753260 -0.10978603
## X9 0.3544268  0.07362113 -0.25111557 -0.13231892 -0.24254817  0.18765482
##            PC7         PC8          PC9
## X1  0.05543869  0.16005914  0.811637429
## X2  0.13657790  0.60411640 -0.048224206
## X3  0.08257828 -0.03476461 -0.094992855
## X4 -0.30849700 -0.39244126 -0.057608736
## X5 -0.67024916 -0.19081879 -0.046144083
## X6 -0.08734948  0.23986950 -0.046794522
## X7  0.03463451 -0.11415807  0.002559605
## X8  0.12889717  0.23397418 -0.567438948
## X9  0.63372357 -0.54081471 -0.016253801

#Printing the summary of the pca to console
summary(pca.df)

## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.6556 0.83917 0.73658 0.43906 0.42420 0.36278
## Proportion of Variance 0.7836 0.07824 0.06028 0.02142 0.01999 0.01462
## Cumulative Proportion  0.7836 0.86182 0.92210 0.94352 0.96352 0.97814
##                            PC7     PC8     PC9
## Standard deviation     0.30315 0.26522 0.18574
## Proportion of Variance 0.01021 0.00782 0.00383
## Cumulative Proportion  0.98835 0.99617 1.00000
```

We get from summary the std deviation, Proportion of Variance and the cummulative variance. If we look at PCA summary we get 92% variance in the first 3 columns. Thus, we can use these 3 variables instead of X1 to X9

Now,Visualizing the results of PCA

```r
#Importing the required libraries
library(factoextra)

## Warning: package 'factoextra' was built under R version 3.5.3

## Loading required package: ggplot2

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at
https://goo.gl/13EFCZ

#Printing the Scree plot of PCA to console
fviz_eig(pca.df)
```



Answer 4.

```r
#Changing levels of Gender to 0,1,2
levels(df$Gender) <- c(0,1,2)

#Changing Gender from ctegorical to numerical
df$Gender <- as.numeric(df$Gender)

#Creating matrix with scaled values
matstd.df <- scale(df[,2:11])

#Applying Kmeans for predicting 5 groups with 5 random points as starting
points
(kmeans5.df <- kmeans(matstd.df,5,nstart = 5))
```

```
## K-means clustering with 5 clusters of sizes 10, 17, 18, 20, 12
##
## Cluster means:
##           X1         X2         X3         X4         X5         X6
## 1 -0.3527640  0.2701040 -0.3882835  0.4163640 -0.4793000  0.1856072
## 2  0.3812756  0.7279403  0.5329382  0.7594103  0.5105006  0.5918743
## 3 -0.2334116 -0.2216926 -0.3487361 -0.1625387 -0.3989820 -0.2496471
## 4 -1.0326917 -1.2801246 -0.7442101 -1.3154104 -0.9210487 -1.2015623
## 5  1.8250998  1.2097445  1.3320283  1.0133575  1.8097616  1.3839132
##            X7         X8          X9      Gender
## 1  0.06722854 -0.3411333  0.09284671  1.87555706
## 2 -0.36415458  0.2825290  0.57799756 -0.26968794
## 3 -0.39078317 -0.2301413 -0.19917961 -0.31259284
## 4 -0.53191468 -0.9427605 -1.25559574 -0.38553117
## 5  1.93256109  1.8005079  1.49522686 -0.06946508
##
## Clustering vector:
##  [1] 3 3 3 3 2 3 3 3 3 3 3 4 3 3 3 2 4 4 4 4 4 4 4 4 3 4 4 4 4 4 4 4 4 4 4
## [36] 4 3 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 3 1 1 1 1 1 1 1 1 1 1 5 5 5 5 5 5 5
## [71] 5 2 5 5 2 5 5
##
## Within cluster sum of squares by cluster:
## [1] 24.97754 38.49080 56.89736 26.74809 39.23482
##  (between_SS / total_SS =  75.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
kmeans5.df$cluster
```

```
##  [1] 3 3 3 3 2 3 3 3 3 3 3 4 3 3 3 2 4 4 4 4 4 4 4 4 3 4 4 4 4 4 4 4 4 4 4
## [36] 4 3 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 3 1 1 1 1 1 1 1 1 1 1 5 5 5 5 5 5 5
## [71] 5 2 5 5 2 5 5
```

```
##Applying Kmeans for predicting 5 groups with 10 random points as starting
points
(kmeans10.df <- kmeans(matstd.df,5,nstart = 10))
```

```
## K-means clustering with 5 clusters of sizes 20, 19, 18, 10, 10
##
## Cluster means:
##           X1         X2         X3         X4         X5         X6
## 1 -1.0326917 -1.2801246 -0.7442101 -1.3154104 -0.9210487 -1.2015623
## 2  0.4465495  0.7255130  0.6108086  0.7108358  0.6049923  0.6375427
## 3 -0.2334116 -0.2216926 -0.3487361 -0.1625387 -0.3989820 -0.2496471
## 4 -0.3527640  0.2701040 -0.3882835  0.4163640 -0.4793000  0.1856072
## 5  1.9898442  1.3107172  1.3438925  1.1564386  1.8900795  1.4555510
##            X7         X8          X9      Gender
```

```
## 1 -0.53191468 -0.9427605 -1.25559574 -0.38553117
## 2 -0.20017854  0.3268102  0.62293299 -0.27420425
## 3 -0.39078317 -0.2301413 -0.19917961 -0.31259284
## 4  0.06722854 -0.3411333  0.09284671  1.87555706
## 5  2.08034976  2.0199694  1.59329540 -0.02083952
##
## Clustering vector:
##  [1] 3 3 3 3 2 3 3 3 3 3 3 1 3 3 3 2 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1
## [36] 1 3 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 3 4 4 4 4 4 4 4 4 4 4 5 5 5 2 5 5 5
## [71] 5 2 5 5 2 2 5
##
## Within cluster sum of squares by cluster:
## [1] 26.74809 49.36404 56.89736 24.97754 28.08717
##  (between_SS / total_SS =  75.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
kmeans10.df$cluster
```

```
##  [1] 3 3 3 3 2 3 3 3 3 3 3 1 3 3 3 2 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1
## [36] 1 3 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 3 4 4 4 4 4 4 4 4 4 4 5 5 5 2 5 5 5
## [71] 5 2 5 5 2 2 5
```

```
##Applying Kmeans for predicting 5 groups with 15 random points as starting
points
(kmeans15.df <- kmeans(matstd.df,5,nstart = 15))
```

```
## K-means clustering with 5 clusters of sizes 10, 20, 19, 10, 18
##
## Cluster means:
##           X1         X2         X3         X4         X5         X6
## 1 -0.3527640  0.2701040 -0.3882835  0.4163640 -0.4793000  0.1856072
## 2 -1.0326917 -1.2801246 -0.7442101 -1.3154104 -0.9210487 -1.2015623
## 3  0.4465495  0.7255130  0.6108086  0.7108358  0.6049923  0.6375427
## 4  1.9898442  1.3107172  1.3438925  1.1564386  1.8900795  1.4555510
## 5 -0.2334116 -0.2216926 -0.3487361 -0.1625387 -0.3989820 -0.2496471
##            X7         X8          X9      Gender
## 1  0.06722854 -0.3411333  0.09284671  1.87555706
## 2 -0.53191468 -0.9427605 -1.25559574 -0.38553117
## 3 -0.20017854  0.3268102  0.62293299 -0.27420425
## 4  2.08034976  2.0199694  1.59329540 -0.02083952
## 5 -0.39078317 -0.2301413 -0.19917961 -0.31259284
##
## Clustering vector:
##  [1] 5 5 5 5 3 5 5 5 5 5 5 2 5 5 5 3 2 2 2 2 2 2 2 2 5 2 2 2 2 2 2 2 2 2 2
## [36] 2 5 3 3 3 3 3 3 3 3 3 3 3 5 5 3 3 5 1 1 1 1 1 1 1 1 1 1 4 4 4 3 4 4 4
## [71] 4 3 4 4 3 3 4
```

```
##
## Within cluster sum of squares by cluster:
## [1] 24.97754 26.74809 49.36404 28.08717 56.89736
##  (between_SS / total_SS =  75.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
kmeans15.df$cluster
```

```
##  [1] 5 5 5 5 3 5 5 5 5 5 5 2 5 5 5 3 2 2 2 2 2 2 2 2 5 2 2 2 2 2 2 2 2 2 2
## [36] 2 5 3 3 3 3 3 3 3 3 3 3 3 5 5 3 3 5 1 1 1 1 1 1 1 1 1 1 4 4 4 3 4 4 4
## [71] 4 3 4 4 3 3 4
```

If we look at the above results we can see that for maximum number of times we get cluster 1 with cluster 5 for datapoints i.e For IndianWolves we are getting close relation ship with Modern Dogs

```
#Converting gender back to factor variable
df$Gender <- as.factor(df$Gender)
```

Answer5:

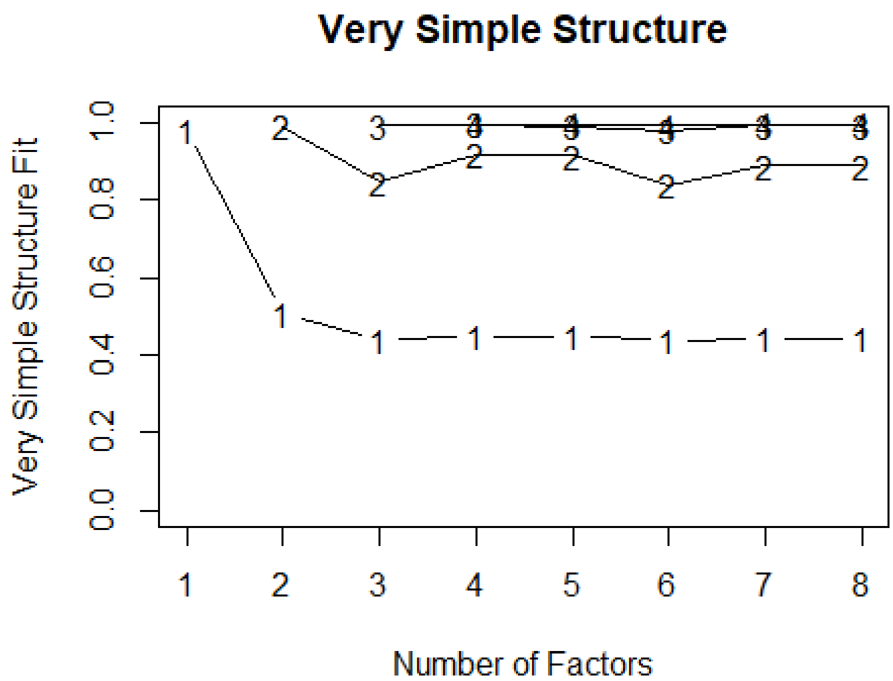
```
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.5.3
```

```
##
## Attaching package: 'psych'
```

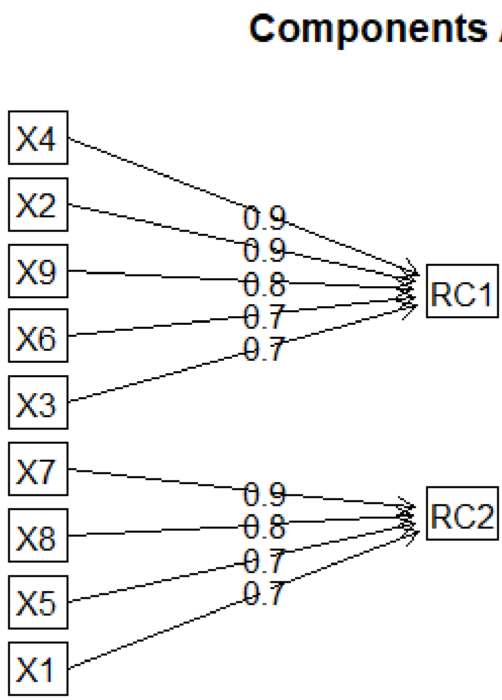
```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
vss(df[,-c(1,11)]) # See factor recommendation
```




```
##
## Very Simple Structure
## Call: vss(x = df[, -c(1, 11)])
## VSS complexity 1 achieves a maximimum of 0.98  with  1  factors
## VSS complexity 2 achieves a maximimum of 0.99  with  2  factors
##
## The Velicer MAP achieves a minimum of 0.08  with  2  factors
## BIC achieves a minimum of  NA  with  2  factors
## Sample Size adjusted BIC achieves a minimum of  NA  with  4  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof   chisq    prob sqresid  fit RMSEA BIC  SABIC
## 1 0.98 0.00 0.094  27 1.4e+02 2.6e-17    0.98 0.98 0.244  23 108.37
## 2 0.51 0.99 0.080  19 4.4e+01 8.6e-04    0.51 0.99 0.141 -38  21.67
## 3 0.44 0.85 0.121  12 1.5e+01 2.2e-01    0.38 0.99 0.072 -37   1.12
## 4 0.45 0.92 0.165   6 4.1e+00 6.6e-01    0.23 1.00 0.000 -22  -3.05
## 5 0.45 0.91 0.174   1 3.0e-01 5.8e-01    0.16 1.00 0.000  -4  -0.89
## 6 0.44 0.84 0.288  -3 4.7e-08      NA    0.13 1.00    NA  NA     NA
## 7 0.44 0.89 0.550  -6 1.4e-06      NA    0.28 0.99    NA  NA     NA
## 8 0.44 0.89 1.000  -8 0.0e+00      NA    0.28 0.99    NA  NA     NA
##   complex  eChisq    SRMR  eCRMS   eBIC
## 1     1.0 1.7e+01 5.5e-02 0.0635 -100.5
## 2     1.7 2.1e+00 2.0e-02 0.0269  -80.4
## 3     2.1 4.1e-01 8.6e-03 0.0150  -51.7
## 4     2.0 9.5e-02 4.1e-03 0.0101  -26.0
## 5     2.0 5.3e-03 9.8e-04 0.0059   -4.3
## 6     2.2 8.6e-10 3.9e-07     NA     NA
```

```
## 7     2.2 1.4e-08 1.6e-06     NA     NA
## 8     2.2 9.9e-18 4.2e-11     NA     NA

# from Vss the recommended factor is 2
pc <- principal(df[,-c(1,11)], nfactors=2, rotate="varimax")
summary(pc)

##
## Factor analysis with Call: principal(r = df[, -c(1, 11)], nfactors = 2,
rotate = "varimax")
##
## Test of the hypothesis that 2 factors are sufficient.
## The degrees of freedom for the model is 19  and the objective function was
1.18
## The number of observations was  77  with Chi Square =  83.28  with prob <
5e-10
##
## The root mean square of the residuals (RMSA) is  0.04

round(pc$values, 3)

## [1] 7.052 0.704 0.543 0.193 0.180 0.132 0.092 0.070 0.035

pc$loadings #From the loadings we can see that upto 2 RC factors explain
about 86% of the variance.

##
## Loadings:
##    RC1   RC2
## X1 0.672 0.700
## X2 0.863 0.390
## X3 0.713 0.255
## X4 0.903 0.278
## X5 0.638 0.701
## X6 0.736 0.553
## X7 0.203 0.932
## X8 0.544 0.783
## X9 0.756 0.564
##
##                  RC1   RC2
## SS loadings    4.375 3.381
## Proportion Var 0.486 0.376
## Cumulative Var 0.486 0.862

pc$communality

##        X1        X2        X3        X4        X5        X6        X7
## 0.9417787 0.8968812 0.5732902 0.8922558 0.8976670 0.8472377 0.9093146
##        X8        X9
## 0.9082632 0.8897015

fa.diagram(pc) #Plotting the EFA Plot
```



Yes, there is relationship between Species and Factor analysis components as these Factors contribute to show common variance in the data.

Answer6:

```
library(MASS)
library(klaR)

## Warning: package 'klaR' was built under R version 3.5.3

library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

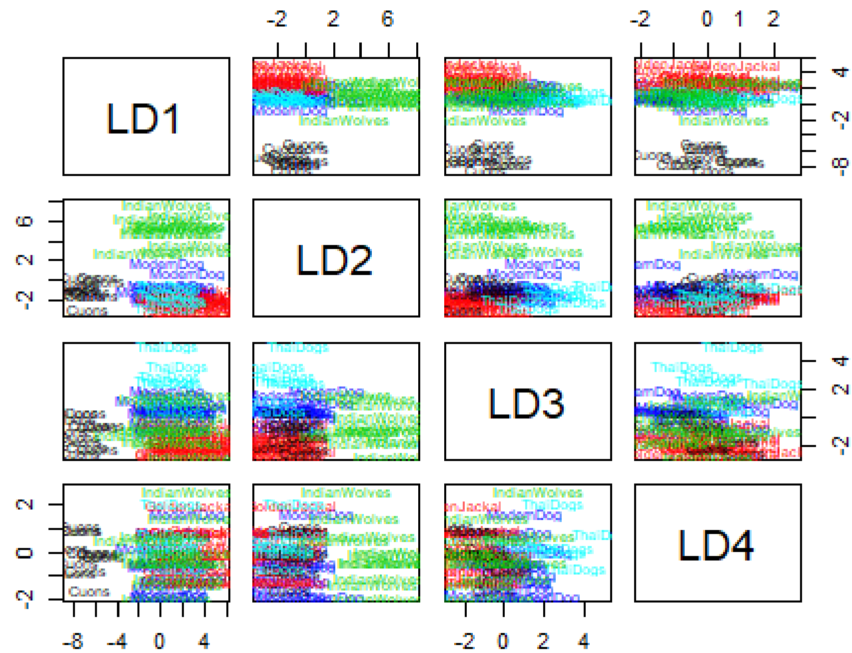
```r
new_data <- df[,-11]
sample_n(new_data, 10)
```

```
##     CanineGroup  X1   X2 X3 X4 X5  X6 X7 X8  X9
## 35 GoldenJackal 111  8.4 17 16 18 7.0 30 34 4.7
## 5     ModernDog 149 12.0 25 25 21 8.4 35 43 6.6
## 6     ModernDog 125  9.5 23 20 20 7.8 33 37 6.3
## 33 GoldenJackal 107  8.4 18 17 18 6.2 29 31 4.3
## 70 IndianWolves 164 12.1 27 24 25 9.9 42 45 8.3
## 47        Cuons 130 11.3 22 23 21 8.7 29 37 7.0
## 30 GoldenJackal 108  8.2 18 16 17 6.5 29 33 4.8
## 32 GoldenJackal 105  8.3 19 17 17 6.5 29 32 4.5
## 19 GoldenJackal 110  8.1 18 16 19 7.1 31 32 4.7
## 57     ThaiDogs 111  9.9 19 20 18 7.3 29 34 5.3
```

```r
training_sample <- sample(c(TRUE, FALSE), nrow(new_data), replace = T, prob =
c(0.8,0.2))
train <- new_data[training_sample, ]
test <- new_data[!training_sample, ]

lda.new_data <- lda(CanineGroup ~ ., train)
lda.new_data #show results
```

```
## Call:
## lda(CanineGroup ~ ., data = train)
##
## Prior probabilities of groups:
##        Cuons GoldenJackal IndianWolves    ModernDog     ThaiDogs
##     0.187500     0.265625     0.187500     0.250000     0.109375
##
## Group means:
##                    X1        X2       X3       X4       X5       X6
## Cuons        134.8333 10.800000 22.41667 23.91667 21.83333 8.566667
## GoldenJackal 111.0000  8.152941 18.58824 17.00000 18.17647 6.817647
## IndianWolves 158.0833 11.733333 26.33333 25.00000 24.83333 9.333333
## ModernDog    125.9375  9.725000 21.37500 21.12500 19.37500 7.675000
## ThaiDogs     119.2857 10.171429 20.00000 22.57143 19.00000 7.900000
##                    X7       X8       X9
## Cuons        29.50000 37.91667 6.700000
## GoldenJackal 30.29412 33.58824 4.847059
## IndianWolves 40.25000 45.16667 7.441667
## ModernDog    32.06250 36.62500 5.868750
## ThaiDogs     32.00000 35.00000 5.971429
##
## Coefficients of linear discriminants:
##            LD1         LD2         LD3         LD4
## X1 -0.04083198  0.01345532 -0.10962653 -0.14366414
## X2  0.17326378  0.14469766  0.60256974 -0.06073781
## X3  0.09580536  0.02204919 -0.02592382 -0.04634062
## X4 -0.30229535 -0.03324196  0.47829162  0.20993021
```

```
## X5 -0.98210964  0.55400828 -1.17304276  0.71048636
## X6 -0.46281304 -0.25618042  0.22394385  0.42602929
## X7  1.16915907  0.34308525  0.33518412  0.11164578
## X8  0.07664643  0.10540910  0.11304778  0.02487802
## X9 -1.51072124 -0.56290388  1.22884815 -0.62822446
##
## Proportion of trace:
##    LD1    LD2    LD3    LD4
## 0.5841 0.3293 0.0818 0.0048

plot(lda.new_data, col = as.integer(train$CanineGroup))
```




```
lda.train <- predict(lda.new_data)
train$lda <- lda.train$class
table(train$lda,train$CanineGroup)

##
##                Cuons GoldenJackal IndianWolves ModernDog ThaiDogs
##   Cuons           12            0            0         0        0
##   GoldenJackal     0           16            0         0        0
##   IndianWolves     0            0           12         0        0
##   ModernDog        0            1            0        16        2
##   ThaiDogs         0            0            0         0        5

lda.test <- predict(lda.new_data,test)
test$lda <- lda.test$class
table(test$lda,test$CanineGroup)
```

```
##
##                Cuons GoldenJackal IndianWolves ModernDog ThaiDogs
##   Cuons            5           0            0         0        0
##   GoldenJackal     0           3            0         0        0
##   IndianWolves     0           0            2         0        0
##   ModernDog        0           0            0         0        1
##   ThaiDogs         0           0            0         0        2
```

Answer7: For Cuons

```
#Extracting dfata for cuons
df.cuons <- df[df$CanineGroup=='Cuons',]

#Dropping CanineGroup column and storing it in a dataframe
df.cuons <- df.cuons[,-1]

#Changing the levels to 0&1 i.e Females and Males
levels(df.cuons$Gender) <- c(0,1,1)

#Applying Logistic regression to with all the variables for Cuons
fit.cuons <- glm(Gender~.,data=df.cuons,family = 'binomial')

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

#Applting stepwise regression to find best variables
final.fit.cuons <- step(fit.cuons)

## Start:  AIC=20
## Gender ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df   Deviance AIC
## - X2    1 9.3930e-10  18
## - X7    1 9.6750e-10  18
## - X5    1 9.9310e-10  18
## - X4    1 9.9390e-10  18
## - X1    1 1.0473e-09  18
## - X8    1 1.0549e-09  18
## - X3    1 2.0975e-09  18
## - X6    1 5.3654e-09  18
## - X9    1 7.2480e-09  18
## <none>    9.3450e-10  20

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=18
## Gender ~ X1 + X3 + X4 + X5 + X6 + X7 + X8 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df   Deviance AIC
## - X7    1 1.0180e-09  16
## - X4    1 1.0206e-09  16
## - X5    1 1.1232e-09  16
## - X1    1 1.4116e-09  16
## - X8    1 1.5474e-09  16
## - X3    1 4.3633e-09  16
## - X9    1 6.9469e-09  16
## - X6    1 1.2524e-08  16
## <none>    9.3930e-10  18

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16
## Gender ~ X1 + X3 + X4 + X5 + X6 + X8 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df Deviance    AIC
## - X4    1    0.000 14.000
## - X1    1    0.000 14.000
## - X8    1    0.000 14.000
## - X5    1    0.000 14.000
## - X3    1    0.000 14.000
## - X9    1    0.000 14.000
## <none>       0.000 16.000
## - X6    1   10.866 24.866

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=14
## Gender ~ X1 + X3 + X5 + X6 + X8 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df Deviance    AIC
## - X1    1   0.0000 12.000
## - X8    1   0.0000 12.000
## - X5    1   0.0000 12.000
## <none>      0.0000 14.000
## - X3    1   7.7103 19.710
## - X9    1  16.6998 28.700
## - X6    1  18.1660 30.166

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=12
## Gender ~ X3 + X5 + X6 + X8 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df Deviance    AIC
## - X8    1   0.0000 10.000
## - X5    1   0.0000 10.000
## <none>      0.0000 12.000
## - X3    1   7.8986 17.899
## - X9    1  17.6290 27.629
## - X6    1  18.1660 28.166

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=10
## Gender ~ X3 + X5 + X6 + X9
##
##        Df Deviance    AIC
## <none>      0.0000 10.000
## - X5    1   7.8475 15.848
## - X3    1   8.0938 16.094
## - X9    1  18.1685 26.169
## - X6    1  18.5302 26.530
```

For cuons we get X3,X5,X6,X9 as most contributing variables to predict gender

For ModernDog

```
#Extracting dfata for cuons
df.moderndog <- df[df$CanineGroup=='ModernDog',]

#Dropping CanineGroup column and storing it in a dataframe
df.moderndog <- df.moderndog[,-1]

#Changing the levels to 0&1 i.e Females and Males
levels(df.moderndog$Gender) <- c(0,1,1)

#Applying Logistic regression to with all the variables for Cuons
fit.moderndog <- glm(Gender~.,data=df.moderndog,family = 'binomial')

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

#Applting stepwise regression to find best variables
final.fit.moderndog <- step(fit.moderndog)

## Start:  AIC=20
## Gender ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##         Df   Deviance AIC
## - X2     1 5.2400e-10  18
## - X3     1 5.4300e-10  18
## - X6     1 5.6300e-10  18
## - X5     1 5.7400e-10  18
## - X9     1 6.9800e-10  18
## - X7     1 9.3200e-10  18
## - X4     1 1.0200e-09  18
## - X8     1 7.3050e-09  18
## - X1     1 3.8625e-08  18
## <none>     4.3800e-10  20

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## 
## Step:  AIC=18
## Gender ~ X1 + X3 + X4 + X5 + X6 + X7 + X8 + X9

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##         Df Deviance   AIC
## - X3    1   0.0000 16.00
## - X6    1   0.0000 16.00
## - X9    1   0.0000 16.00
## - X5    1   0.0000 16.00
## - X7    1   0.0000 16.00
## - X4    1   0.0000 16.00
## - X8    1   0.0000 16.00
## <none>      0.0000 18.00
## - X1    1   9.8201 25.82

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16
## Gender ~ X1 + X4 + X5 + X6 + X7 + X8 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##        Df Deviance    AIC
## - X6    1   0.0000 14.000
## - X5    1   0.0000 14.000
## - X9    1   0.0000 14.000
## - X4    1   0.0000 14.000
## - X7    1   0.0000 14.000
## <none>      0.0000 16.000
## - X8    1   8.8363 22.836
## - X1    1  10.2749 24.275

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=14
## Gender ~ X1 + X4 + X5 + X7 + X8 + X9

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df Deviance    AIC
## - X5    1   0.0000 12.000
## - X9    1   0.0000 12.000
## <none>      0.0000 14.000
## - X4    1   7.1863 19.186
## - X8    1   9.2476 21.248
## - X1    1  11.1937 23.194
## - X7    1  14.5018 26.502

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=12
## Gender ~ X1 + X4 + X7 + X8 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df Deviance    AIC
## - X9    1   0.0000 10.000
## <none>      0.0000 12.000
```

```
## - X4    1   9.4321 19.432
## - X8    1   9.4832 19.483
## - X1    1  11.3406 21.341
## - X7    1  16.3343 26.334

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=10
## Gender ~ X1 + X4 + X7 + X8
##
##        Df Deviance    AIC
## <none>      0.0000 10.000
## - X4    1   9.9914 17.991
## - X8    1  11.2905 19.291
## - X1    1  12.8696 20.870
## - X7    1  18.1140 26.114
```

For ModernDog we get variables X4,X8,X1,X7 as most contributing variables

For GoldenJackal

```
#Extracting dfata for cuons
df.GoldenJackal <- df[df$CanineGroup=='GoldenJackal',]

#Dropping CanineGroup column and storing it in a dataframe
df.GoldenJackal <- df.GoldenJackal[,-1]

#Changing the levels to 0&1 i.e Females and Males
levels(df.GoldenJackal$Gender) <- c(0,1,1)

#Applying Logistic regression to with all the variables for Cuons
fit.GoldenJackal <- glm(Gender~.,data=df.GoldenJackal,family = 'binomial')

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

#Applting stepwise regression to find best variables
final.fit.GoldenJackal <- step(fit.GoldenJackal)

## Start:  AIC=20
## Gender ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##         Df   Deviance AIC
## - X3     1 1.1501e-09  18
## - X8     1 1.1747e-09  18
## - X4     1 1.1758e-09  18
## - X9     1 1.3420e-09  18
## - X5     1 1.3552e-09  18
## - X6     1 1.5406e-09  18
## - X1     1 1.7927e-09  18
## - X7     1 1.8378e-09  18
## - X2     1 4.2338e-09  18
## <none>     1.1415e-09  20

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=18
## Gender ~ X1 + X2 + X4 + X5 + X6 + X7 + X8 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df   Deviance AIC
## - X4    1 1.2020e-09  16
## - X8    1 1.4551e-09  16
## - X6    1 1.5372e-09  16
## - X5    1 1.8172e-09  16
## - X7    1 1.8482e-09  16
## - X1    1 2.6216e-09  16
## - X9    1 3.0477e-09  16
## - X2    1 3.8965e-09  16
## <none>    1.1501e-09  18

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16
## Gender ~ X1 + X2 + X5 + X6 + X7 + X8 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df   Deviance AIC
## - X8    1 1.4356e-09  14
## - X7    1 1.7422e-09  14
## - X5    1 2.3021e-09  14
## - X9    1 3.6386e-09  14
## - X6    1 4.0799e-09  14
## - X1    1 1.0388e-08  14
## - X2    1 1.2225e-08  14
## <none>    1.2020e-09  16

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=14
## Gender ~ X1 + X2 + X5 + X6 + X7 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df Deviance    AIC
## - X7    1   0.0000 12.000
## - X5    1   0.0000 12.000
## - X9    1   0.0000 12.000
## - X2    1   0.0000 12.000
## - X1    1   0.0000 12.000
## <none>      0.0000 14.000
## - X6    1   5.6182 17.618

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=12
## Gender ~ X1 + X2 + X5 + X6 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df Deviance    AIC
## - X5    1   0.0000 10.000
## - X9    1   0.0000 10.000
## - X2    1   0.0000 10.000
## - X1    1   0.0000 10.000
## <none>      0.0000 12.000
## - X6    1   5.6261 15.626

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Step:  AIC=10
## Gender ~ X1 + X2 + X6 + X9

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df Deviance    AIC
## - X2    1   0.0000  8.000
## - X1    1   0.0000  8.000
## <none>      0.0000 10.000
## - X9    1   7.7888 15.789
## - X6    1   8.4675 16.468

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=8
## Gender ~ X1 + X6 + X9
##
##        Df Deviance    AIC
## <none>      0.0000  8.000
## - X1    1   7.8698 13.870
## - X6    1   8.6179 14.618
## - X9    1   9.8447 15.845
```

From the above results we can say that for GoldenJackal we have X1,X6,X9 as he most contributing variables

For IndianWolves

```
#Extracting dfata for cuons
df.IndianWolves <- df[df$CanineGroup=='IndianWolves',]

#Dropping CanineGroup column and storing it in a dataframe
df.IndianWolves <- df.IndianWolves[,-1]

#Changing the levels to 0&1 i.e Females and Males
levels(df.IndianWolves$Gender) <- c(0,1,1)

#Applying Logistic regression to with all the variables for Cuons
fit.IndianWolves <- glm(Gender~.,data=df.IndianWolves,family = 'binomial')

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
#Applting stepwise regression to find best variables
final.fit.IndianWolves <- step(fit.IndianWolves)

## Start:  AIC=20
## Gender ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df   Deviance AIC
## - X9    1 2.3819e-10  18
## - X2    1 2.3871e-10  18
## - X4    1 3.8639e-10  18
## - X6    1 4.8390e-10  18
## - X3    1 5.0125e-10  18
## - X1    1 5.0833e-10  18
## - X5    1 5.3413e-10  18
## - X8    1 5.4460e-10  18
## - X7    1 5.7355e-10  18
## <none>    3.4651e-10  20

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## 
## Step:  AIC=18
## Gender ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df   Deviance AIC
## - X4    1 2.7584e-10  16
## - X8    1 4.2524e-10  16
## - X6    1 4.4206e-10  16
## - X1    1 4.8572e-10  16
## - X3    1 5.1832e-10  16
## - X7    1 6.2639e-10  16
## - X5    1 7.1771e-10  16
## - X2    1 1.6137e-09  16
## <none>    2.3819e-10  18

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16
## Gender ~ X1 + X2 + X3 + X5 + X6 + X7 + X8

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df   Deviance AIC
## - X6    1 4.9580e-10  14
## - X8    1 6.0980e-10  14
## - X3    1 6.9290e-10  14
## - X1    1 6.9320e-10  14
## - X7    1 8.1330e-10  14
## - X5    1 1.5389e-09  14
## - X2    1 3.4172e-09  14
## <none>    2.7580e-10  16

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=14
## Gender ~ X1 + X2 + X3 + X5 + X7 + X8

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##        Df   Deviance AIC
## - X8    1 6.8620e-10  12
## - X3    1 7.5510e-10  12
## - X1    1 7.6540e-10  12
## - X7    1 8.8060e-10  12
## - X5    1 1.7412e-09  12
## - X2    1 5.0559e-09  12
## <none>    4.9580e-10  14
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=12
## Gender ~ X1 + X2 + X3 + X5 + X7

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##         Df Deviance    AIC
## - X1    1    0.000 10.000
## - X3    1    0.000 10.000
## - X7    1    0.000 10.000
## - X5    1    0.000 10.000
## <none>       0.000 12.000
## - X2    1   10.146 20.146

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=10
## Gender ~ X2 + X3 + X5 + X7

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##         Df Deviance    AIC
## - X3    1    0.000  8.000
## - X7    1    0.000  8.000
## - X5    1    0.000  8.000
## <none>       0.000 10.000
## - X2    1   10.727 18.727

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=8
## Gender ~ X2 + X5 + X7
##
##         Df Deviance    AIC
## <none>      0.0000  8.000
## - X7    1   5.3755 11.376
## - X5    1   8.0172 14.017
## - X2    1  11.7552 17.755
```

For IndianWolves we get most contributing factors as X7,X5,X2.

If we analyze the above results we get most frequent variables as X1,X5,X6,X7,X9 as the contributing variables

Hence we can fit a logistic regression with variables X1,X5,X6,X7,X9,X2

```
#Verifying the above results
df.all.except.thaidogs <- df[!(df$CanineGroup =='ThaiDogs'),]

#Dropping CanineGroup column and storing it in a dataframe
df.all.except.thaidogs <- df.all.except.thaidogs[,-1]

#Changing the levels to 0&1 i.e Females and Males
levels(df.all.except.thaidogs$Gender) <- c(0,1,1)

#Applying Logistic regression to with all the variables for Cuons
fit.all.except.thaidogs <- glm(Gender~.,data=df.all.except.thaidogs,family =
'binomial')

#Applting stepwise regression to find best variables
final.fit.all.except.thaidogs <- step(fit.all.except.thaidogs)

## Start:  AIC=101.94
## Gender ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9
##
##         Df Deviance    AIC
## - X5    1   82.144 100.14
## - X7    1   82.258 100.26
## - X1    1   82.452 100.45
```

```
## - X6    1   82.741 100.74
## - X8    1   83.076 101.08
## - X4    1   83.082 101.08
## - X3    1   83.130 101.13
## - X2    1   83.696 101.70
## - X9    1   83.781 101.78
## <none>      81.939 101.94
##
## Step:  AIC=100.14
## Gender ~ X1 + X2 + X3 + X4 + X6 + X7 + X8 + X9
##
##        Df Deviance     AIC
## - X7    1   82.336  98.336
## - X1    1   82.682  98.682
## - X4    1   83.308  99.308
## - X3    1   83.322  99.322
## - X8    1   83.386  99.386
## - X6    1   83.579  99.579
## - X9    1   83.798  99.798
## - X2    1   83.837  99.837
## <none>      82.144 100.144
##
## Step:  AIC=98.34
## Gender ~ X1 + X2 + X3 + X4 + X6 + X8 + X9
##
##        Df Deviance    AIC
## - X1    1   83.049 97.049
## - X8    1   83.398 97.398
## - X3    1   83.564 97.564
## - X6    1   83.603 97.603
## - X9    1   83.856 97.856
## - X2    1   84.170 98.170
## <none>      82.336 98.336
## - X4    1   84.480 98.480
##
## Step:  AIC=97.05
## Gender ~ X2 + X3 + X4 + X6 + X8 + X9
##
##        Df Deviance    AIC
## - X8    1   83.400 95.400
## - X6    1   83.919 95.919
## - X9    1   84.563 96.563
## - X4    1   84.719 96.719
## - X2    1   84.871 96.871
## - X3    1   84.937 96.937
## <none>      83.049 97.049
##
## Step:  AIC=95.4
## Gender ~ X2 + X3 + X4 + X6 + X9
##
```

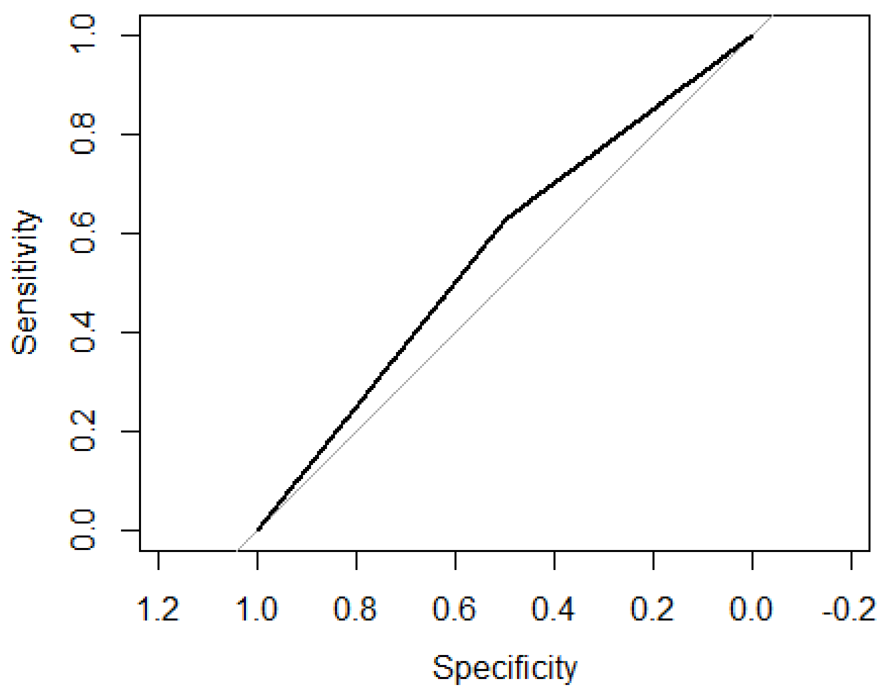
```
##          Df Deviance    AIC
## - X9     1   84.578 94.578
## - X6     1   84.634 94.634
## - X3     1   85.094 95.094
## - X4     1   85.171 95.171
## <none>       83.400 95.400
## - X2     1   85.456 95.456
##
## Step:  AIC=94.58
## Gender ~ X2 + X3 + X4 + X6
##
##          Df Deviance    AIC
## - X6     1   84.936 92.936
## - X3     1   86.121 94.121
## - X4     1   86.297 94.297
## <none>       84.578 94.578
## - X2     1   88.241 96.241
##
## Step:  AIC=92.94
## Gender ~ X2 + X3 + X4
##
##          Df Deviance    AIC
## - X3     1   86.415 92.415
## <none>       84.936 92.936
## - X4     1   87.201 93.201
## - X2     1   88.358 94.358
##
## Step:  AIC=92.41
## Gender ~ X2 + X4
##
##          Df Deviance    AIC
## - X4     1   87.911 91.911
## <none>       86.415 92.415
## - X2     1   90.416 94.416
##
## Step:  AIC=91.91
## Gender ~ X2
##
##          Df Deviance    AIC
## <none>       87.911 91.911
## - X2     1   92.747 94.747
```

From above results we get that X2 is the contributing variable. Thus we fit the new model with minimum number of parameters

Answer9:

```
#Training the model with entire data and parameters X1,X5,X6,X7,X9,X2
df.all.except.thaidogs <- df[!(df$CanineGroup =='ThaiDogs'),]
```

```r
#Dropping CanineGroup column and storing it in a dataframe
df.all.except.thaidogs <- df.all.except.thaidogs[,-1]

#Extracting the relevant columns
df.all.except.thaidogs <-
df.all.except.thaidogs[,c('X1','X5','X6','X7','X9','X2','Gender')]


#Changing the levels to 0&1 i.e Females and Males
levels(df.all.except.thaidogs$Gender) <- c(0,1,1)

#Applying Logistic regression to with all the variables
fit.all.except.thaidogs <- glm(Gender~.,data=df.all.except.thaidogs,family =
'binomial')

#Calculating the accuracy of Logistic Regression
cm <-
table(df.all.except.thaidogs$Gender,as.factor(ifelse(test=as.numeric(fit.all.
except.thaidogs$fitted.values>0.5) == 0, yes=1, no=0)))
accuracy <-  (cm[1,1] + cm[2,2]) / (cm[1,1] + cm[2,2] + cm[1,2] + cm[2,1])
accuracy

## [1] 0.4328358

library(pROC)

## Warning: package 'pROC' was built under R version 3.5.3

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

roc(df.all.except.thaidogs$Gender,ifelse(test=as.numeric(fit.all.except.thaid
ogs$fitted.values>0.5) == 0, yes=1, no=0),plot=TRUE)
```




```
##
## Call:
## roc.default(response = df.all.except.thaidogs$Gender, predictor =
ifelse(test = as.numeric(fit.all.except.thaidogs$fitted.values >     0.5) ==
0, yes = 1, no = 0), plot = TRUE)
##
## Data: ifelse(test = as.numeric(fit.all.except.thaidogs$fitted.values >
0.5) == 0, yes = 1, no = 0) in 32 controls (df.all.except.thaidogs$Gender 0)
> 35 cases (df.all.except.thaidogs$Gender 1).
## Area under the curve: 0.5643
```

Answer8: Area under curve is very less hence, the model has low accuracy.

Answer9a: We used Logistic Regression becsause we had binary outcome so Logistic Regression Answer9b: We get an accuracy of 43.28% for Logistic Regression

```
#Creating test data
df.test <- df[(df$CanineGroup
=='ThaiDogs'),c('X1','X5','X6','X7','X9','X2','Gender')]

#Predicting the values
predicted_values <- predict(fit.all.except.thaidogs,newdata = df.test[,-
c(7)])

#Predicting the values of Male & Female
predicted_values <- as.factor(ifelse(test=as.numeric(predicted_values>0.5) ==
0, yes="Male", no="Female"))
```

```r
#Printing Values to console
predicted_values

##  [1] Female Female Female Male   Female Female Male   Female Female Male
## Levels: Female Male

#Replacing all the values by predicted values
df[df$CanineGroup=='ThaiDogs',]$Gender<-
as.factor(ifelse(test=as.numeric(predicted_values>0.5) == 0, yes=2, no=1))

## Warning in Ops.factor(predicted_values, 0.5): '>' not meaningful for
## factors
```

Answer10: We have to create a linear regression model to predict Mandible Length i.e X1

```r
#Extracting al the data except for Thaidogs
names(df)

##  [1] "CanineGroup" "X1"          "X2"          "X3"          "X4"
##  [6] "X5"          "X6"          "X7"          "X8"          "X9"
## [11] "Gender"

df.all.except.thaidogs <- df[!(df$CanineGroup =='ThaiDogs'),c(2:10)]

#Creating the model with X1 against all the variables
fit.lm <- lm(X1~.,data = df.all.except.thaidogs)

#Printing the summary of the model
summary(fit.lm)

##
## Call:
## lm(formula = X1 ~ ., data = df.all.except.thaidogs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0282  -1.9666   0.2624   2.6000  10.2278
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.6009     4.9450  -2.548  0.01350 *
## X2            0.4696     1.0314   0.455  0.65061
## X3            0.3451     0.1542   2.238  0.02911 *
## X4            1.1334     0.4196   2.701  0.00904 **
## X5            0.3042     0.6434   0.473  0.63813
## X6            0.1986     1.3123   0.151  0.88022
## X7            0.4763     0.2299   2.072  0.04273 *
## X8            1.9582     0.2897   6.759 7.44e-09 ***
## X9            1.5098     1.4110   1.070  0.28905
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.979 on 58 degrees of freedom
## Multiple R-squared:  0.9586, Adjusted R-squared:  0.9529
## F-statistic:   168 on 8 and 58 DF,  p-value: < 2.2e-16
```

From the above summary we can say that we get an accuracy of around 95% if we look at adjusted R-squared value.

Now we predict the values for Thai Dogs

```
#Creating test data
df.test.thaidogs <- df[(df$CanineGroup =='ThaiDogs'),c(2:10)]


#Predicting the values for X!
predicted_values <- predict(fit.lm,newdata =df.test.thaidogs[,-c(1)])


#Loading the required library
library(ggplot2)

#Plotting the predicted values with actual values
qplot(df.test.thaidogs$X1, predicted_values) + geom_abline(intercept=0,
slope=1)
```