Week 2

Abhishek Soalnki

11/5/2020

```
library(data.table)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

library(ggplot2)
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.0 --

## v tibble  3.0.1     v purrr   0.3.3
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

The train dataset downloaded from Kaggle is loaded here

```
# Loading dataset
case_data <- read.csv(file = "C:/Users/abhis/Documents/R
Studio/ML/train.csv", header = TRUE)
```

Once the dataset is loaded, I'm going to do very preliminary exploration of the dataset

```
names(case_data)

##   [1] "ID_code" "target"  "var_0"   "var_1"   "var_2"   "var_3"   "var_4"  
##   [8] "var_5"   "var_6"   "var_7"   "var_8"   "var_9"   "var_10"  "var_11" 
##  [15] "var_12"  "var_13"  "var_14"  "var_15"  "var_16"  "var_17"  "var_18" 
##  [22] "var_19"  "var_20"  "var_21"  "var_22"  "var_23"  "var_24"  "var_25" 
##  [29] "var_26"  "var_27"  "var_28"  "var_29"  "var_30"  "var_31"  "var_32" 
##  [36] "var_33"  "var_34"  "var_35"  "var_36"  "var_37"  "var_38"  "var_39" 
##  [43] "var_40"  "var_41"  "var_42"  "var_43"  "var_44"  "var_45"  "var_46" 
##  [50] "var_47"  "var_48"  "var_49"  "var_50"  "var_51"  "var_52"  "var_53" 
##  [57] "var_54"  "var_55"  "var_56"  "var_57"  "var_58"  "var_59"  "var_60" 
##  [64] "var_61"  "var_62"  "var_63"  "var_64"  "var_65"  "var_66"  "var_67" 
##  [71] "var_68"  "var_69"  "var_70"  "var_71"  "var_72"  "var_73"  "var_74" 
##  [78] "var_75"  "var_76"  "var_77"  "var_78"  "var_79"  "var_80"  "var_81" 
##  [85] "var_82"  "var_83"  "var_84"  "var_85"  "var_86"  "var_87"  "var_88" 
##  [92] "var_89"  "var_90"  "var_91"  "var_92"  "var_93"  "var_94"  "var_95" 
##  [99] "var_96"  "var_97"  "var_98"  "var_99"  "var_100" "var_101"
"var_102"
## [106] "var_103" "var_104" "var_105" "var_106" "var_107" "var_108"
"var_109"
## [113] "var_110" "var_111" "var_112" "var_113" "var_114" "var_115"
"var_116"
## [120] "var_117" "var_118" "var_119" "var_120" "var_121" "var_122"
"var_123"
## [127] "var_124" "var_125" "var_126" "var_127" "var_128" "var_129"
"var_130"
## [134] "var_131" "var_132" "var_133" "var_134" "var_135" "var_136"
"var_137"
## [141] "var_138" "var_139" "var_140" "var_141" "var_142" "var_143"
"var_144"
## [148] "var_145" "var_146" "var_147" "var_148" "var_149" "var_150"
"var_151"
## [155] "var_152" "var_153" "var_154" "var_155" "var_156" "var_157"
"var_158"
## [162] "var_159" "var_160" "var_161" "var_162" "var_163" "var_164"
"var_165"
## [169] "var_166" "var_167" "var_168" "var_169" "var_170" "var_171"
"var_172"
## [176] "var_173" "var_174" "var_175" "var_176" "var_177" "var_178"
"var_179"
## [183] "var_180" "var_181" "var_182" "var_183" "var_184" "var_185"
"var_186"
## [190] "var_187" "var_188" "var_189" "var_190" "var_191" "var_192"
"var_193"
## [197] "var_194" "var_195" "var_196" "var_197" "var_198" "var_199"
```

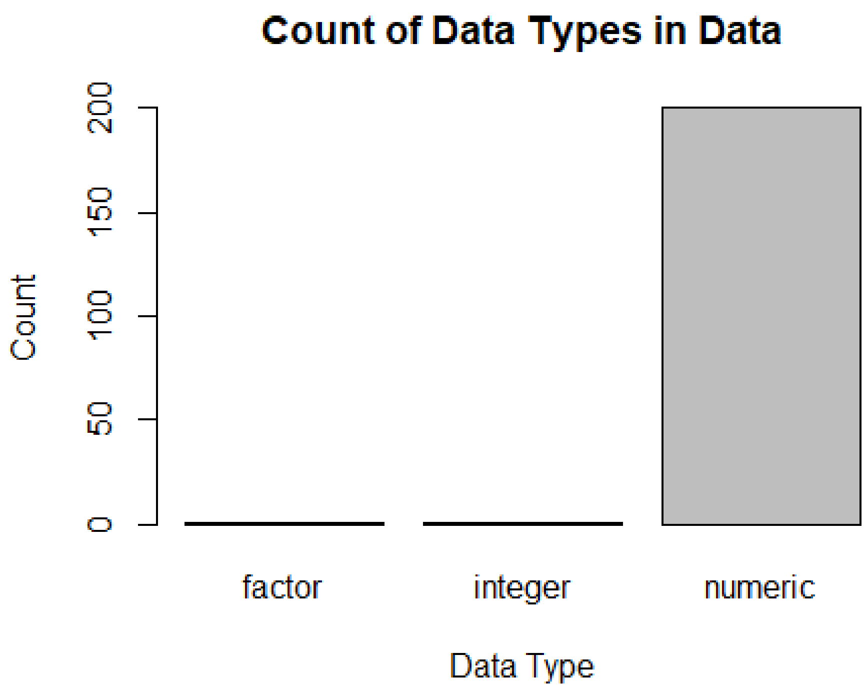
It is clear that this is a very big dataset. In fact, there are total of 202 variables in the dataset and except the first two variables, non of the others are provided with any headers to understand the column or find any relation between thew column for further analysis.

ID code is the unique identifier variable and as such does not provide any valuable information to the dataset and thus the variable can be simply ignored.

Target variable is the response variable in the dataset and it is a binary response variable.

Now let us explore the type of variables present in the dataset:

```
barplot(table(data.frame(sapply(case_data, class))),
        ylab = 'Count', xlab = 'Data Type', main = 'Count of Data Types in
Data')
```



The dataset clearly displays that 200 of the 202 variables are numeric which is making the variables even more anonymous since there are no headers and the numerical variables can mean anything. If there were a few categorical variables it would be easier to determine any relation and explore the dataset.

ID Code variable we discussed earlier is the only character data type and it is not an informative variable. The only interger data type would be the 'target' variable since it is a binary variable thus would always be an integer.

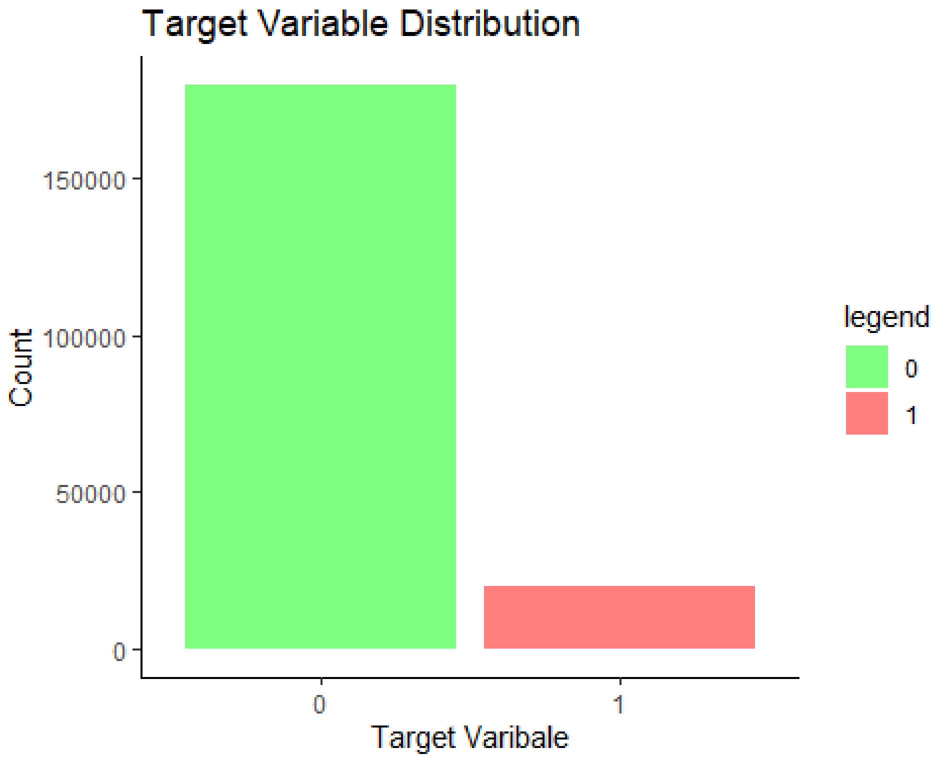
Let us check if there are null values in the dataset:

```
sum(is.na(case_data))

## [1] 0
```

There are no missing values in the dataset which is great. This helps us insure that the quality of data would not be compromised due to missing value treatment.

The distribution of the 2 classes in the dataset would further help us understand the quality of the data and whether we have encountered a case of unbalanced classification problem or not.

```
target_df <- data.frame(table(case_data$target))
colnames(target_df) <- c("target", "count")
ggplot(data=target_df, aes(x=target, y=count, fill=target)) +
  geom_bar(position = 'dodge', stat='identity', alpha=0.5) +
  scale_fill_manual("legend", values = c("1" = "red", "0"="green")) +
  labs(x="Target Varibale", y= "Count", title ="Target Variable
Distribution")+
  theme_classic()
```



The target value distribution shows unbalanced class scenario as the count values for both the responses are highly skewed.

89.95% of the data has target response value as ‘0’ and 10.05% of the target values are ‘1’.

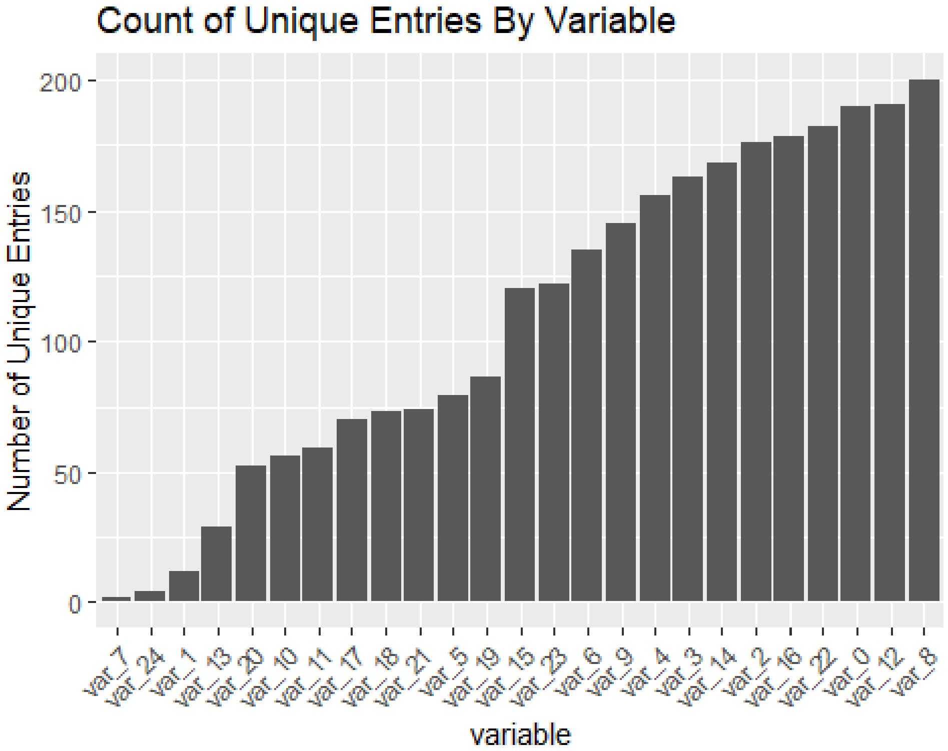
```
unique.count <- t(case_data[,3:202] %>% summarise_all(n_distinct))
unique.count <- cbind(newColName = rownames(unique.count), unique.count)
rownames(unique.count) <- 1:nrow(unique.count)
colnames(unique.count) <- c("Variable", "Unique")
unique.count <- data.frame(unique.count)
unique.count$Unique <- as.integer(unique.count$Unique)

summary(unique.count$Unique)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    1.00   50.75  100.50  100.50  150.25  200.00
```

From the data we observe that the maximum number of unique entries observed are 200 in any variable and we also have variables that contain same value for all the 200000 rows.

Plotting data for first 25 variables:

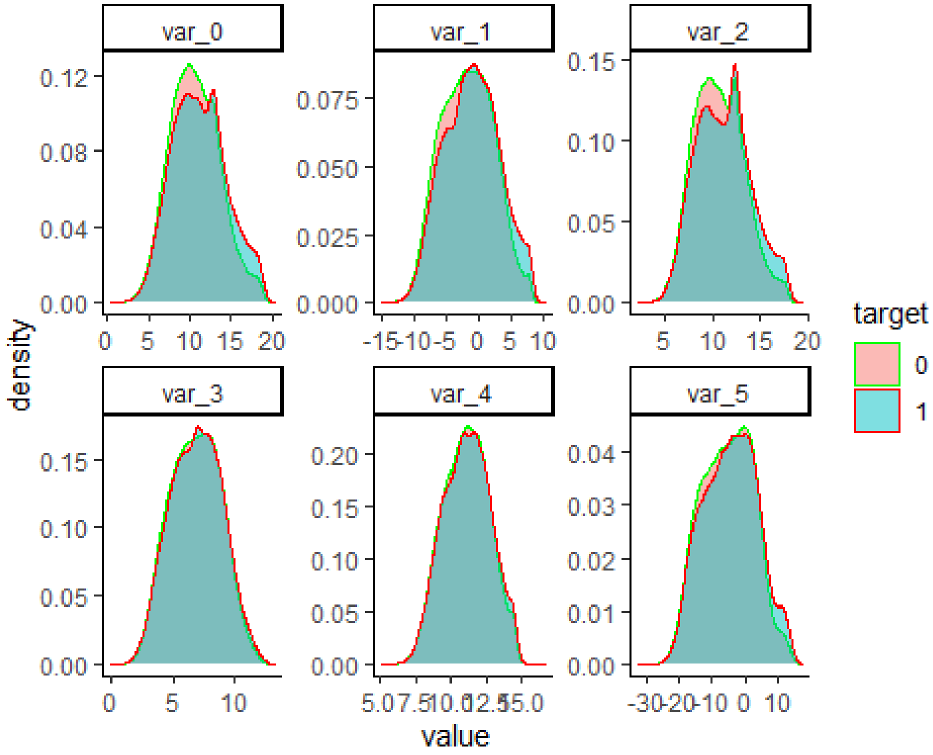
```
unique_25 <- unique.count[1:25,]
unique_25$Variable <- factor(unique_25$Variable, levels = 
unique_25$Variable[order(unique_25$Unique)])

ggplot(data = unique_25) +
    geom_bar(aes(x=Variable, y=Unique), stat = 'identity') +
    labs(x='variable', y="Number of Unique Entries", title='Count of Unique 
Entries By Variable') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



We see that the maximum number of unique entries are for Variable 8. This provides us with the insight that most of the data values are repeated in the dataset of 200000 rows.

Let us visualize the density plot for few numeric variables:

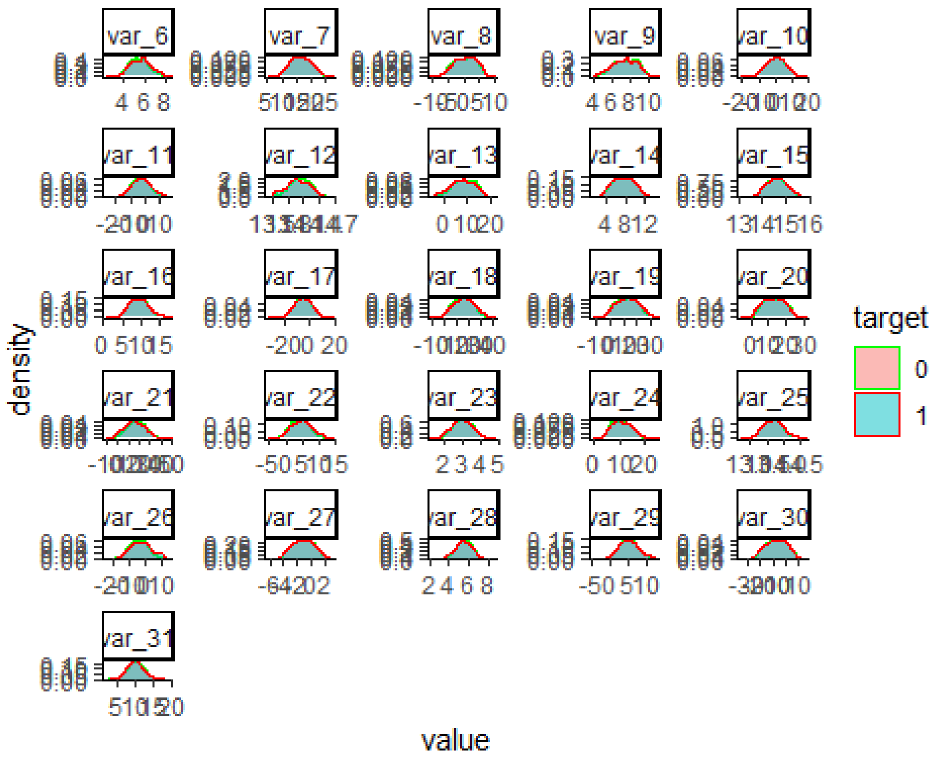
```
feature_groups <- 3:8
col_names <- colnames(case_data)[c(2,feature_groups)]
temp <- gather(case_data[,col_names], key="features", value="value", -target)
temp$target <- factor(temp$target)
temp$features <- factor(temp$features, levels=col_names[-1],
labels=col_names[-1])
ggplot(data=temp, aes(x=value)) +
  geom_density(aes(fill=target, color=target), alpha=0.5) +
  scale_color_manual(values = c("1" = "red", "0"="green")) +
  theme_classic() +
  facet_wrap(~ features, ncol = 3, scales = "free")
```



From the density plot for first 6 numeric variables, it is somewhat observed that there is not significant difference in the values on comparing density plots by target value. The plots have very large overlap area which clearly helps us conclude that atleast for the first 6 variables, we do not see any significantly different values under the 2 target groups.

We take advantage of facet in ggplot and develop density plot for next 25 variables:

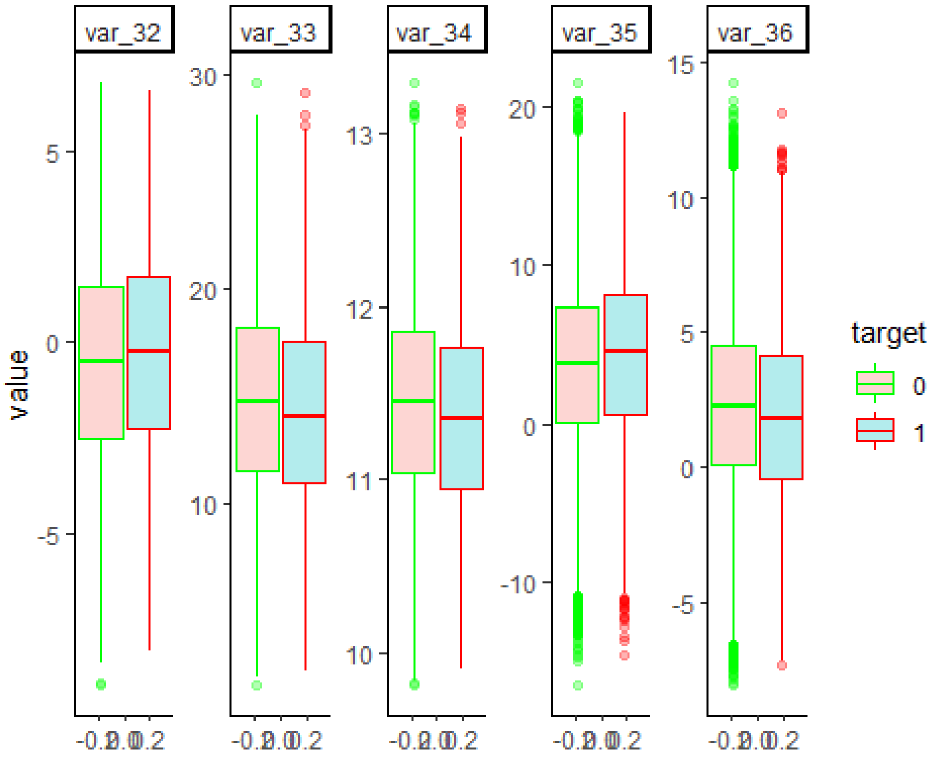
```
feature_groups <- 9:34
col_names <- colnames(case_data)[c(2,feature_groups)]
temp <- gather(case_data[,col_names], key="features", value="value", -target)
temp$target <- factor(temp$target)
temp$features <- factor(temp$features, levels=col_names[-1],
labels=col_names[-1])
ggplot(data=temp, aes(x=value)) +
  geom_density(aes(fill=target, color=target), alpha=0.5) +
  scale_color_manual(values = c("1" = "red", "0"="green")) +
  theme_classic() +
  facet_wrap(~ features, ncol = 5, scales = "free")
```



We see a slight variation in density plots for variable 12, 13, 21 and 26. Apart from these, it seems like the values for variables till 'var_31' do not have significantly different distribution for target group 0 and 1.

Boxplots are another good way of comparing the variation in the data and we will compare the boxplots

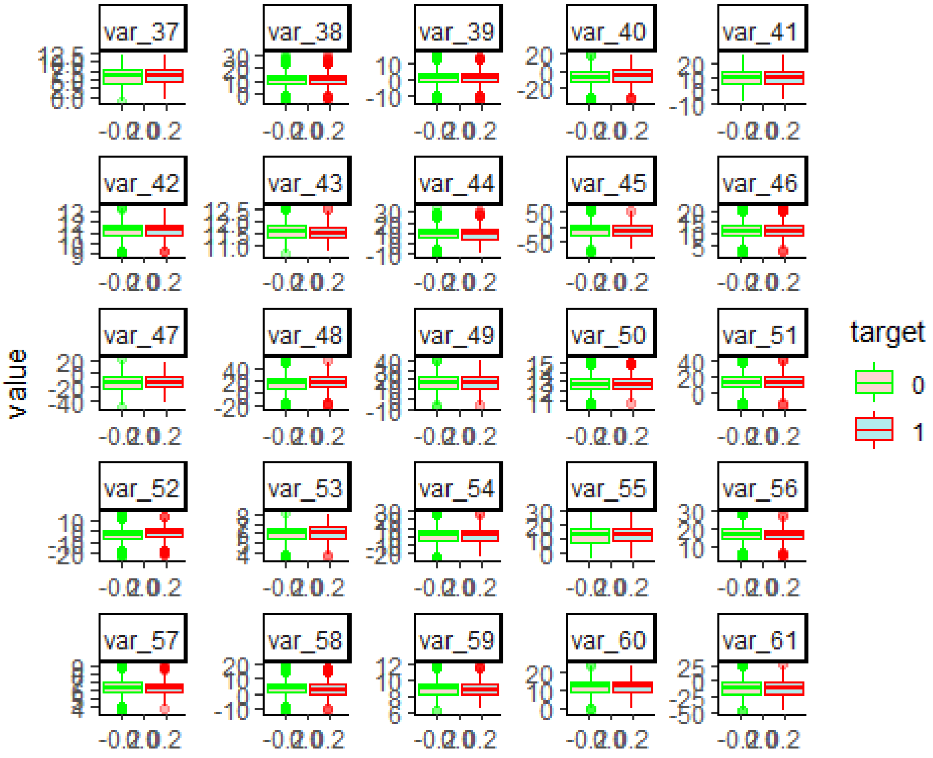
```
feature_groups <- 35:39
col_names <- colnames(case_data)[c(2,feature_groups)]
temp <- gather(case_data[,col_names], key="features", value="value", -target)
temp$target <- factor(temp$target)
temp$features <- factor(temp$features, levels=col_names[-1],
labels=col_names[-1])
ggplot(data=temp, aes(y=value)) +
  geom_boxplot(aes(fill=target, color=target), alpha=0.3) +
  scale_color_manual(values = c("1" = "red", "0"="green")) +
  theme_classic() +
  facet_wrap(~ features, ncol = 5, scales = "free")
```



From the boxplot we observe that the variation in numeric values for target group 0 and 1 is very similar as the Inter Quantile Range (IQR) values are close by. Also the median values are comparable within the target group as well as for the 5 variables from 32 to 36 which gives a belief that all the variables seem to be coming for a similar underlying population distribution.

Next, we plot boxplot for several more variables to compare how the median value and variation follow for these variables:

```r
feature_groups <- 40:64
col_names <- colnames(case_data)[c(2,feature_groups)]
temp <- gather(case_data[,col_names], key="features", value="value", -target)
temp$target <- factor(temp$target)
temp$features <- factor(temp$features, levels=col_names[-1],
labels=col_names[-1])
ggplot(data=temp, aes(y=value)) +
  geom_boxplot(aes(fill=target, color=target), alpha=0.3) +
  scale_color_manual(values = c("1" = "red", "0"="green")) +
  theme_classic() +
  facet_wrap(~ features, ncol = 5, scales = "free")
```

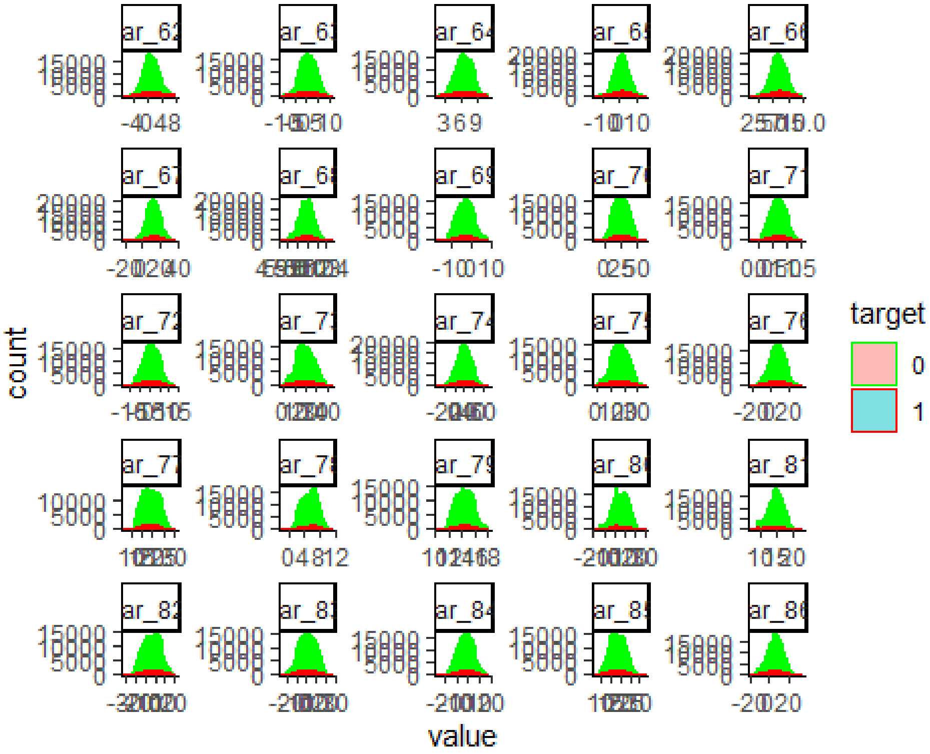


As was our intuition, the median values within target groups are very similar and so is the variation which makes us believe clearly that all of the variables are following similar distribution.

This leads us to the conclusion that either the variables contain random values from a fixed distribution or all the data is already standardized in the training set.

Let us plot histogram for a few variables to strengthen our belief that the variables are standardized because in that case, we should see a normal approximation for our variables:

```
feature_groups <- 65:89
col_names <- colnames(case_data)[c(2,feature_groups)]
temp <- gather(case_data[,col_names], key="features", value="value", -target)
temp$target <- factor(temp$target)
temp$features <- factor(temp$features, levels=col_names[-1],
labels=col_names[-1])
ggplot(data=temp, aes(x=value)) +
  geom_histogram(aes(fill=target, color=target), alpha=0.5) +
  scale_color_manual(values = c("1" = "red", "0"="green")) +
  theme_classic() +
  facet_wrap(~ features, ncol = 5, scales = "free")
```



Yes, our belief is correct and the data in the training set is already standardized.

Next, we want to understand the relationship between different variables in the dataset. There are 200 numeric variables and we want to see if there is high correaltion amoong the variables. For this we calculate the pearson correlation for all the variables and generate a 200 x 200 matrix of correlated values.

It is difficult to visualize a 200x200 plot and therefore we generate the 5-number summary of the values to understand the correlation values:

```
cormat <- cor(case_data[,-c(1,2)], method = c("pearson"))
summary(cormat[upper.tri(cormat)])

##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -9.844e-03 -1.640e-03  1.808e-05  2.781e-05  1.723e-03  9.714e-03
```

The minimum and maximum values are so small and are approximately 0 which gives us insight that none of the 2 variables out of the 200 in the data are correlated.

Overall, the conclusion is that the dataset is unbalanced with ~90% of the data following under one of the target class i.e. 0. We also know that all the 200 predictor variables in the dataset are numerical variables and most of the variables have repeated values since 200 is the maximum number of unique entries in any variable. Overall there does not seem to be a clear distinction in data distribution for the 2 target groups on comparing the density plots, boxplots as well as histograms.