ADVANCE ANALYTICS AND MACHINE LEARNING
ASSIGNMENT-1

STUDENT NAME- SIDDHARTH SHEKHAR SINGH

STUDENT ID-40425947

MSc – FINANCIAL ANALYTICS

INTRODUCTION

The assignment delves into advanced analytics and machine learning, focusing on several aspects of data analysis using the R programming language. It begins with the setup and importation of necessary libraries, such as SmartEDA, tidyverse, glmnet, and others, which are fundamental for data manipulation, visualization, and modeling. The dataset is loaded and processed, including handling of character columns and summarization of missing values.

The core of the assignment revolves around statistical modeling and machine learning techniques. It includes:

- Train-test splitting for model validation.
- Lasso regression to identify critical predictors and manage dimensionality.
- Logistic regression for binary outcome prediction.
- KNN (K-nearest neighbours) for classification tasks.
- Cross-validation for assessing model reliability.

For each method, detailed implementation steps are followed by analysis and interpretation of the results. The assignment aims to uncover insights into product issues consequences through exploratory data analysis, graphical data exploration, and predictive modelling. It concludes with evaluations of model performances and comparisons between different approaches to identify the most suitable predictive models for the dataset at hand.

This thorough exploration encompasses data pre-processing, exploratory analysis, feature selection, and the application of various predictive models to gain insights into the underlying patterns of the dataset, highlighting the power of statistical learning in understanding and predicting outcomes based on historical data.

Importing the libraries

```
library(SmartEDA)

## Warning: package 'SmartEDA' was built under R version 4.3.3

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

library(tidyverse)

## Warning: package 'ggplot2' was built under R version 4.3.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
```

```
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.5.0     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
```

```
## ── Conflicts ─────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force
all conflicts to become errors
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.3.3
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```
library(caTools)
library(MASS)
```

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

```
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(dplyr)
library(pROC)

## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

library(kableExtra)

## Warning: package 'kableExtra' was built under R version 4.3.3

##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows

library(caret)

## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift

library(viridis)

## Loading required package: viridisLite

library(ggplot2)
```

#1-Loading the dataset

```
# Upload the data
data <- read_csv(file.choose())

## Rows: 50646 Columns: 77
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr   (4): ID_non_uniq, Proudct.issue.consequence,
product.field_description...
## dbl  (72): last_year_all_product_codes_num_uniq,
last_year_all_product_codes...
## date  (1): date_event
```

```
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
# To Display the data
head(data)
```

```
## # A tibble: 6 × 77
##   ID_non_uniq date_event last_year_all_product_codes_nu…¹
last_year_all_produc…²
##   <chr>       <date>                                <dbl>
<dbl>
## 1 p890057     2013-04-22                                2
1968
## 2 p890057     2013-04-22                                2
1962
## 3 p890057     2013-04-22                                2
1964
## 4 p890057     2013-04-22                                1
1967
## 5 p890057     2013-04-22                                1
1976
## 6 p080032     2013-05-28                                1
5542
## # ℹ abbreviated names: ¹last_year_all_product_codes_num_uniq,
## #   ²last_year_all_product_codes_most_freq
## # ℹ 73 more variables: last_year_brand_name_num_uniq <dbl>,
## #   last_year_brand_name_most_freq <dbl>,
## #   last_year_classification0_num_uniq <dbl>,
## #   last_year_classification1_num_uniq <dbl>,
## #   last_year_classification2_num_uniq <dbl>, …
```

```
# Removed the ID column
data <- subset(data, select = -c(ID_non_uniq))
```

```
print("Success - Data Loaded !!")
```

```
## [1] "Success - Data Loaded !!"
```

#REQUIRED FUNCTIONS

###Train - Test Split

```
train_test_split <- function(sub){
  set.seed(40425947)

  # Generate random indices for splitting
  split_indices <- sample(nrow(sub), size = floor(0.7 * nrow(sub)), replace =
FALSE)

  # Split data based on indices
  train <- sub[split_indices, ]
  test <- sub[-split_indices, ]

  ### Splitting the train-test data into x_train, y_train, x_test, and
y_test.
  x_train <- model.matrix(Issue_Consequence_new ~ . - 1, data = train)
  y_train <- ifelse(train$Issue_Consequence_new == "Injury", 1, 0)

  x_test <- model.matrix(Issue_Consequence_new ~ . - 1, data = test)
  y_test <- ifelse(test$Issue_Consequence_new == "Injury",1,0)

  train$Issue_Consequence_new <- as.factor(ifelse(train$Issue_Consequence_new
== "Injury", 1, 0))
  test$Issue_Consequence_new <- as.factor(ifelse(test$Issue_Consequence_new
== "Injury", 1, 0))

  return(list(X_train = x_train, y_train = y_train, x_test = x_test, y_test =
y_test, train = train, test = test))
}
```

LASSO REGRESSION MODEL

```
lasso <- function(x_train, y_train, x_test, y_test, subsets){


  grid = 10^seq(10, -2, length = 100)
  cvmodel_A01 <-  cv.glmnet(x_train, y_train, alpha = 1 , family =
"binomial", type.measure = "deviance", nfolds = 10)

  plot_cvmodel <- plot(cvmodel_A01)
  plot_cvmodel

  best_lambda <- cvmodel_A01$lambda.min
  best_lambda

  # fitting lasso model on train set
  set.seed(40425947)
  lasso.mod <- glmnet(x_train, y_train,family="binomial",type.measure =
```

```
"deviance", alpha=1, lambda=grid)
  plot(lasso.mod, label=TRUE)
  plot(lasso.mod, xvar="lambda", label=TRUE, lwd=6)

  plot_lasso.mod <- plot(lasso.mod)
  plot_lasso.mod

  set.seed(40425947)
  lasso.pred <- predict(lasso.mod,family="binomial", s = best_lambda, newx =
x_test)
  pred_class <- ifelse(lasso.pred >= 0.5, 1, 0)


  accuracy <- mean(pred_class == as.numeric(y_test))
  accuracy

  set.seed(40425947)
  X <- subsets %>% dplyr::select(-c(Issue_Consequence_new))
  y <- subsets$Issue_Consequence_new


  out = glmnet(X, y, alpha=1, family="binomial", type.measure = "deviance",
lambda = grid)
  out

  set.seed(40425947)
  lasso.coef <-  predict(out, type="coefficients", s=best_lambda)[1:76,]
  lc <- lasso.coef[lasso.coef!=0]

  lc_df <- data.frame(variable = names(lc)[-1], coefficient = lc[-1])
  print(lc_df)

  top10 <- head(lc_df[order(abs(lc_df$coefficient), decreasing = TRUE), ],
10)
  top10

  plot_top_10 <- ggplot(lc_df, aes(x = variable, y = coefficient)) +
    geom_bar(stat = "identity", fill = "steelblue") +
    theme(axis.text.x = element_text(size=6, angle = 90, vjust = 0.5,
hjust=1)) +
    xlab("Variable") + ylab("Coefficient")

  plot_top_10

  return(list(out = out,
              plot_cvmodel= plot_cvmodel,best_lambda =
best_lambda,plot_lasso.mod =plot_lasso.mod,accuracy = accuracy,top10 =
top10,plot_top_10 =  plot_top_10,lc_df = lc_df))
```

```
}
```

##LOGISTIC REGRESSION

```
logistic_Regression <- function(formula,train, test, y_test){

    glm.fits <- glm(formula, data = train, family = "binomial")
    summary(glm.fits)

    # prediction on test dataset using trained model
    predictions <- predict.glm(glm.fits, test,family = "binomial",
type="response")

    # changing numeric values of predictions into factor ;> 0.5 as 1 and <0.5
as 0
    class_predict <- as.factor(ifelse(predictions > 0.5, 1, 0))
    class_predict

    accuracy_logistic_regression <- mean(class_predict == y_test)
    cat("The Accuracy of the Logistic Regression is :",
accuracy_logistic_regression * 100)

    # predicted probabilities
    pred.prob <- ifelse(fitted(glm.fits) > 0.5, 1, 0)
    head(data.frame(class_predict, test$Issue_Consequence_new))

    # Analysing the residuals
    # Examining the influential cases
    # Examining Multicollinearity

    vif(glm.fits)

    outcome_lr <- postResample(class_predict, as.factor(y_test))
    acc_lr <- outcome_lr["Accuracy"]
    acc_lr

    # Confusion Matrix - to check True/false positive/negatives]
    cm_lr <- confusionMatrix(data=class_predict, as.factor(y_test))
    cm_lr

    cat("The accuracy of Logistic Regression model is", acc_lr, "\n")
```

```
    # Plot of confusion matrix
    plot(cm_lr$table, col = c("white", "blue"),
         main = paste("Confusion Matrix\n Logistic Regression"))


    # extracting coeficients of LR model
    coef_summary <- summary(glm.fits)$coef
    kable(coef_summary, caption = "Logistic Regression Coefficients",
          align = "c", booktabs = TRUE)


    # ROC Curve
    roc_lr <- roc(response = as.numeric(y_test), predictor =
as.numeric(class_predict))
    plot(roc_lr, main = "ROC Curve of LR", col = "blue", print.auc = TRUE,
legacy.axes = TRUE)




}
```

KNN

```
knn_classification <- function(train_data, test_data, formula, k_values) {
  # Define train control
  trControl <- trainControl(method = "cv", number = 10)

  # Fit KNN model
  set.seed(40425947)
  fit_knn <- train(formula,
                   data = train_data,
                   method = "knn",
                   trControl = trControl,
                   tuneGrid = expand.grid(k = k_values),
                   preProcess = c("center", "scale"),
                   metric = "Accuracy")

  # Predictions on test data
  predictions_knn <- predict(fit_knn, newdata = test_data)

  # Post-Resample Accuracy
  acc_knn <- postResample(test_data$Issue_Consequence_new,
predictions_knn)["Accuracy"]

  # Confusion Matrix
  cm_knn <- confusionMatrix(test_data$Issue_Consequence_new, predictions_knn)
```

```
  # Plot of confusion matrix
  plot(cm_knn$table, col = c("white", "blue"),
       main = paste("Confusion Matrix\n KNN"))

  # ROC Curve
  roc_knn <- roc(test_data$Issue_Consequence_new,
as.numeric(predictions_knn))
  plot(roc_knn, main = "ROC Curve of KNN", col = "blue", print.auc = TRUE,
legacy.axes = TRUE)

  return(list(model = fit_knn,
              post_resample_accuracy = acc_knn,
              confusion_matrix = cm_knn,
              roc_curve = roc_knn))
}
```

#Kfold

```
# K-fold value here is 10
trControl <- trainControl(method  = "cv",number  = 10)


# fitting LR with different k values 1:10
kfold_lr <- function(subset){
  fit_lr <- train(as.factor(Issue_Consequence_new)~.,
              method     = "glm",
              tuneGrid   = NULL,
              trControl  = trControl,
              data       =  data,
              metric = "Accuracy")


  return(fit_lr$results)
}
```

########################REPORT##########################

#2 Summarizing Blank and Zero Values in a Dataset

```
# Load the necessary library
library(dplyr)

# Function to summarize NA and 0 values for each column
summarize_blanks_zeros <- function(df) {
  summary_df <- df %>% summarise_all(~ sum(is.na(.) | . == 0))
  return(summary_df)
}

# Apply the function to dataset
summary_blanks_zeros <- summarize_blanks_zeros(data)

# View the summary
print(summary_blanks_zeros)
```

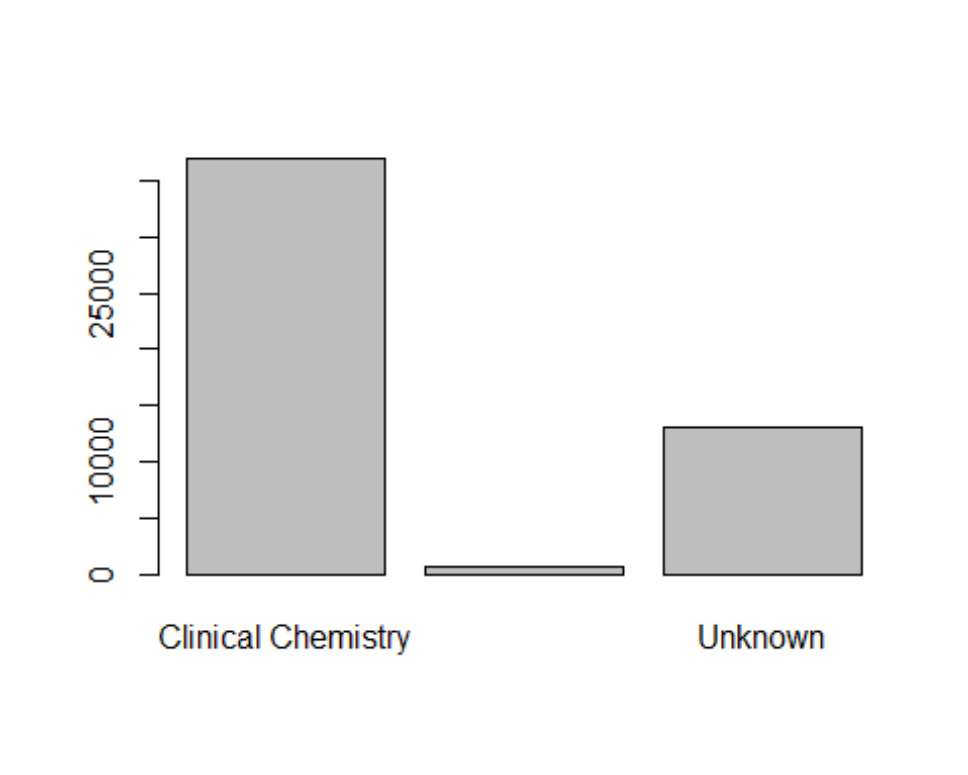
```
## # A tibble: 1 × 76
##   date_event last_year_all_product_codes_num_uniq
last_year_all_product_codes_…¹
##        <int>                                <int>
<int>
## 1          0                                11669
11669
## # ℹ abbreviated name: ¹last_year_all_product_codes_most_freq
## # ℹ 73 more variables: last_year_brand_name_num_uniq <int>,
## #   last_year_brand_name_most_freq <int>,
## #   last_year_classification0_num_uniq <int>,
## #   last_year_classification1_num_uniq <int>,
## #   last_year_classification2_num_uniq <int>,
## #   last_year_company_name_num_uniq <int>, …
```

#3"Converting Character Columns to Factors in Dataframe"

```
data <- data %>% mutate_if(sapply(data, is.character), as.factor)
```

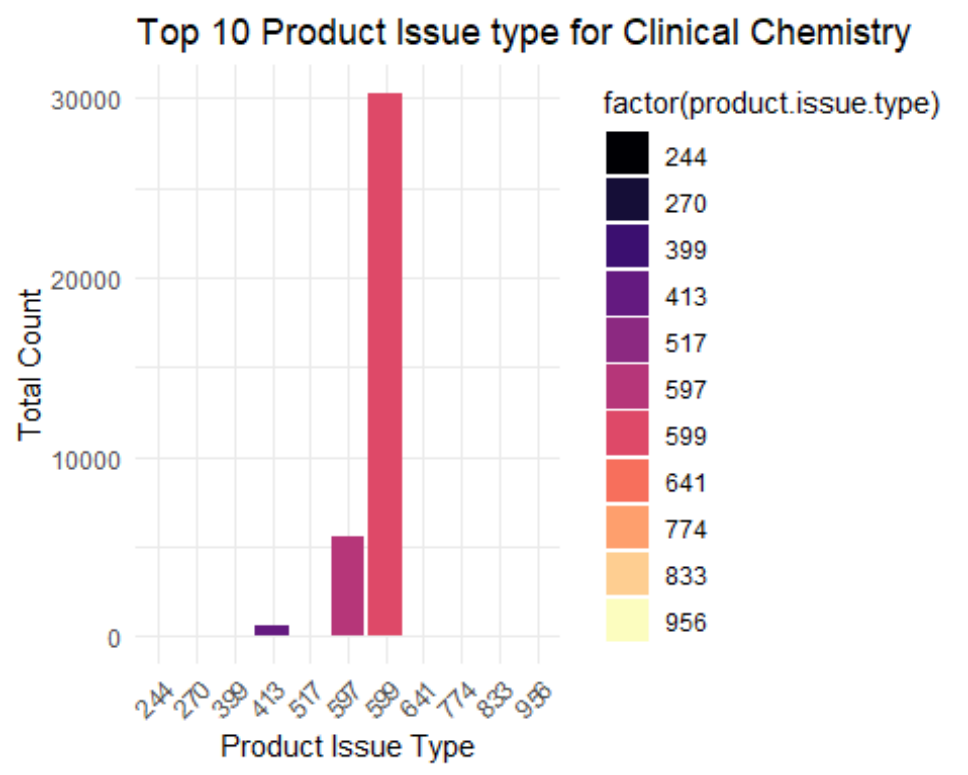
#GRAPHICAL EXPLORATION OF product.field_description and top preditors

```
t <- table(data$product.field_description)

barplot(t)
```

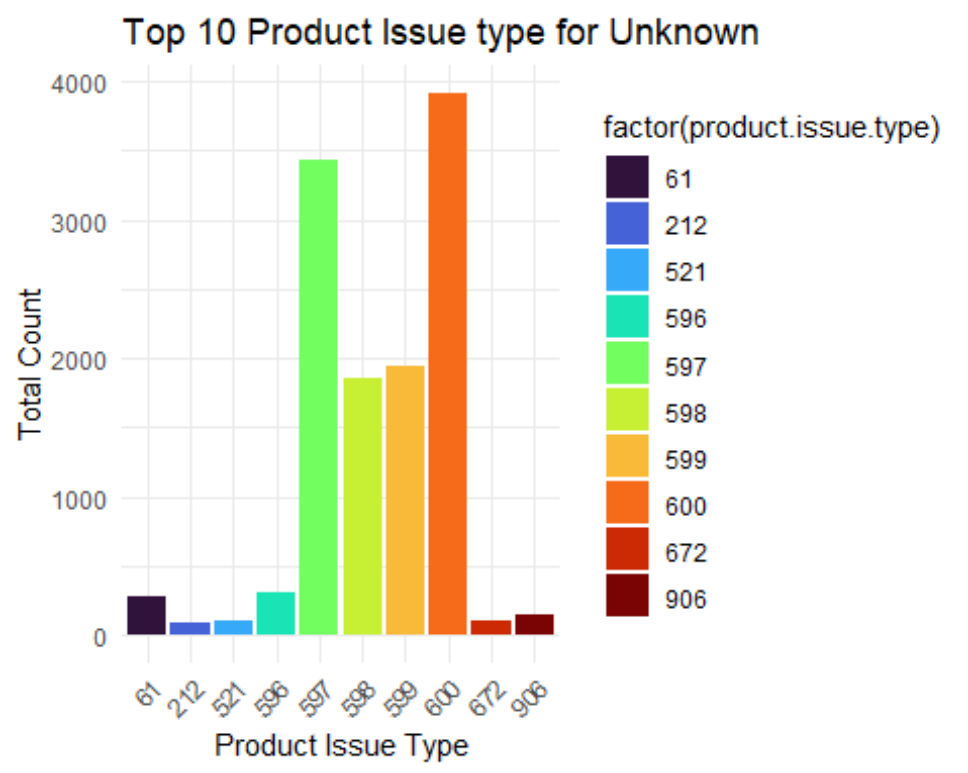



```
filtered_dataset<- data %>%
  group_by(product.field_description, product.issue.type) %>%
  summarise(total_counts = n()) %>%
  arrange(product.field_description, desc(total_counts)) %>%
  mutate(rank = rank(desc(total_counts), ties.method = "min")) %>%
  arrange(product.field_description, rank) %>% filter(rank <= 10)

## `summarise()` has grouped output by 'product.field_description'. You can
## override using the `.groups` argument.

Clinical_Chemistry <- filtered_dataset %>% filter(product.field_description
== "Clinical Chemistry")

unknown <- filtered_dataset %>% filter(product.field_description == "Unknown"
)

General_Hospital <- filtered_dataset %>% filter(product.field_description ==
"General Hospital" )


Clinical_Chemistry_barplot <- ggplot(Clinical_Chemistry, aes(x =
factor(product.issue.type), y = total_counts, fill =
factor(product.issue.type))) +
  geom_bar(stat = "identity") +
  labs(x = "Product Issue Type", y = "Total Count" , title = "Top 10 Product
Issue type for Clinical Chemistry") +
  theme_minimal() +
```

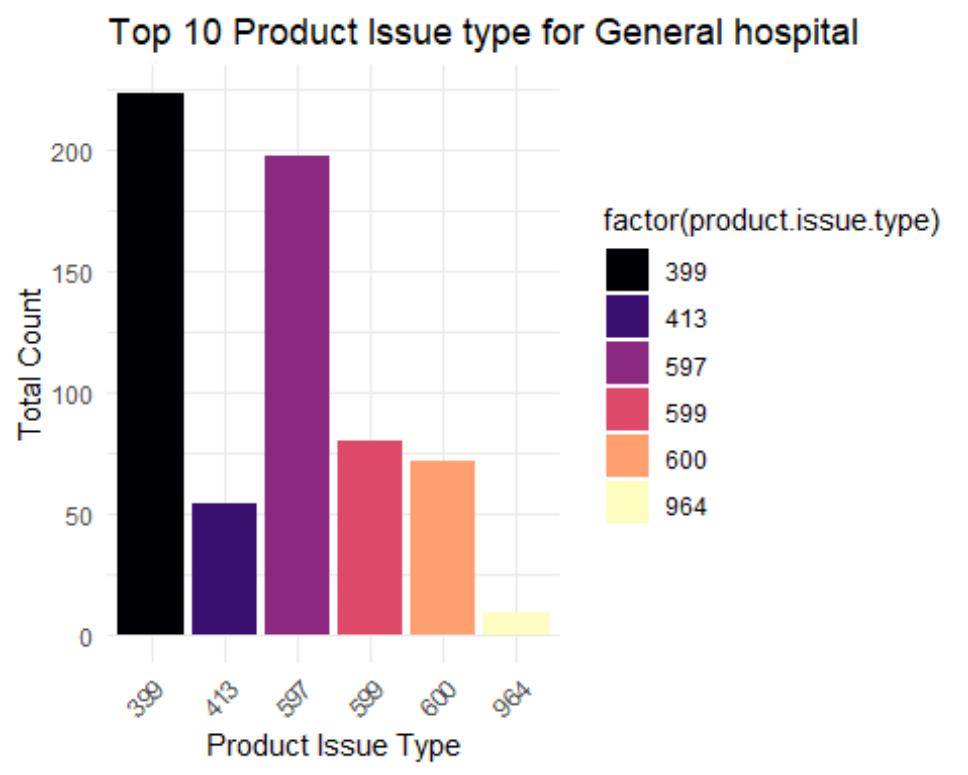
```
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_viridis(discrete = T, option = "A")
Clinical_Chemistry_barplot
```




```
unkown_barplot <- ggplot(unknown, aes(x = factor(product.issue.type), y =
total_counts, fill = factor(product.issue.type))) +
  geom_bar(stat = "identity") +
  labs(x = "Product Issue Type", y = "Total Count", title = "Top 10 Product
Issue type for Unknown") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_viridis(discrete = T, option = "H")

unkown_barplot
```




```
General_Hospital_barplot <- ggplot(General_Hospital, aes(x =
factor(product.issue.type), y = total_counts, fill =
factor(product.issue.type))) +
  geom_bar(stat = "identity") +
  labs(x = "Product Issue Type", y = "Total Count" , title = "Top 10 Product
Issue type for General hospital") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_viridis(discrete = T, option = "A")
General_Hospital_barplot
```



INSIGHTS FROM PRODUCT FIELD DESCRIPTION

The bar plots generated for the top 10 product issue types within the product fields of Clinical Chemistry, Unknown, and General Hospital offer valuable insights into the distribution and frequency of product issues in these distinct areas. Here's an analysis of the insights derived from these visualizations: Clinical Chemistry Insight:For Clinical Chemistry, the bar plots show a range of issue types with different counts, indicating varying frequencies of issues within this field. Notably, some issue types (like those labeled 774, 833, and 956) have much higher counts, which could point to more common problems within this field. This might suggest that certain chemical products are more prone to issues or that there is a pattern in manufacturing defects or contamination in this category.

General Hospital Insight: Like Clinical Chemistry, the General Hospital field shows a range of issue frequencies but with lower overall counts. This may reflect a smaller volume of products or possibly more effective quality assurance practices in products designed for hospital settings.

Unknown Insight:The Unknown category also presents a range of issue types, but the counts are generally lower than those in Clinical Chemistry. This suggests that while there may be a variety of issues present, they occur with less frequency. This could imply better quality control or fewer reporting of issues, possibly due to less direct impact on consumer health or less stringent monitoring.

The insights drawn from these bar plots not only shed light on the specific product issues prevalent in different fields but also highlight the importance of clear product categorization and detailed issue reporting. By focusing on fields with direct implications

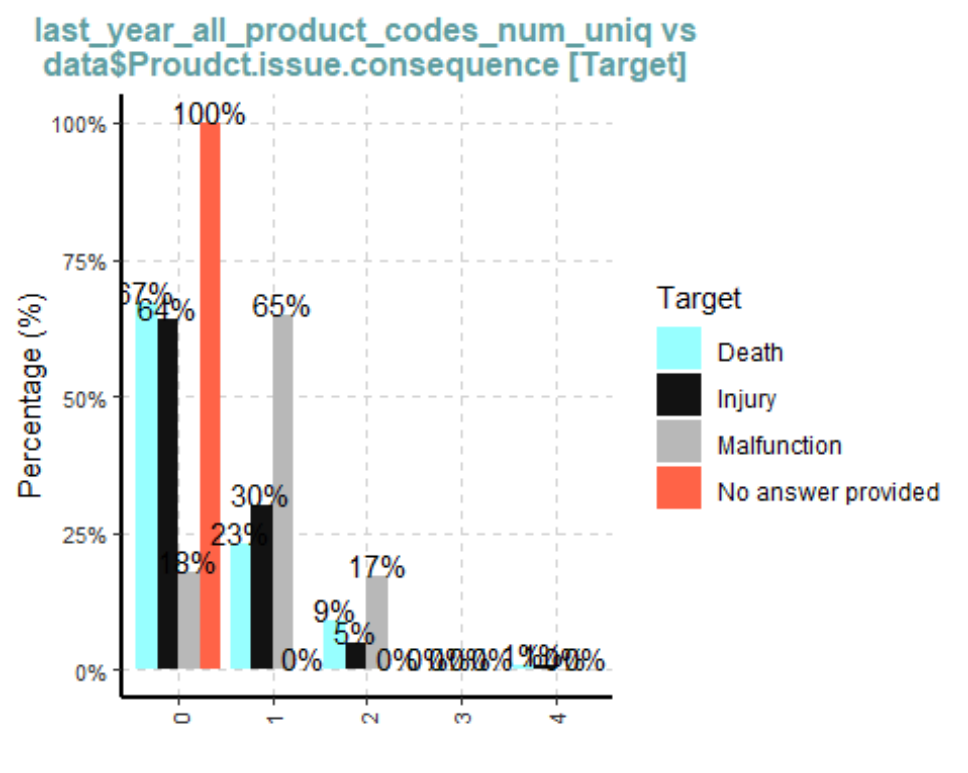
for patient care and safety, this analysis underlines the critical need for robust quality assurance and regulatory oversight in healthcare-related products.

Column Generation for the Descriptive Statistics

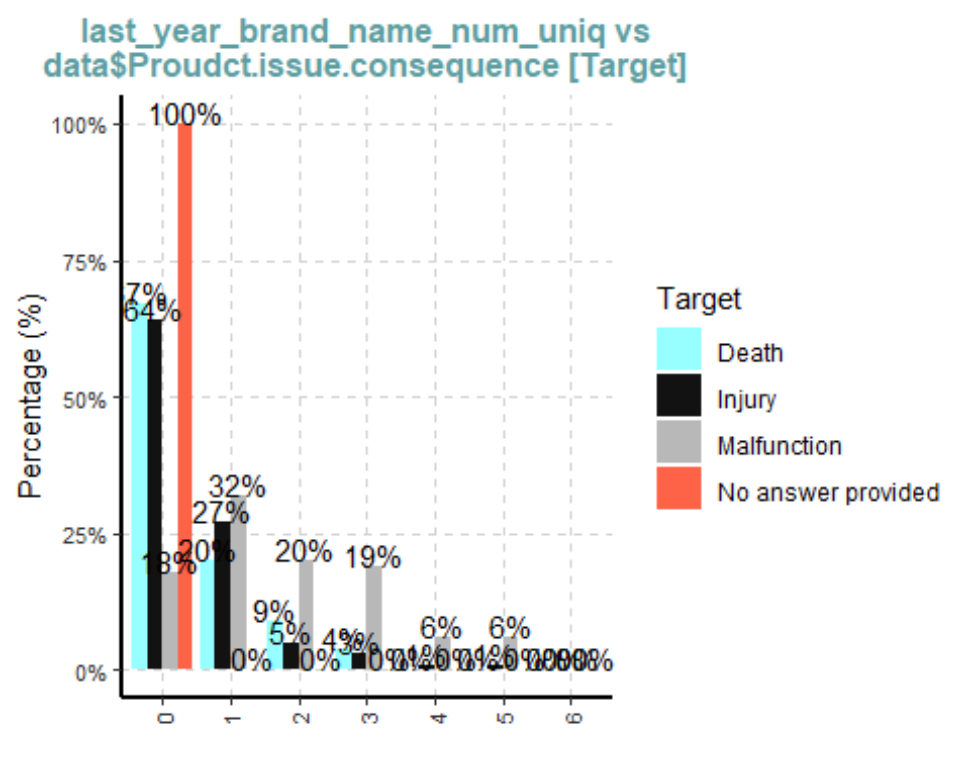
```
last_year <- data %>% dplyr::select(starts_with("last_year"))
last_two_year <- data %>% dplyr::select(starts_with("last_two_year"))
last_four_year <- data  %>% dplyr::select(starts_with("last_four_year"))
product <- data %>% dplyr::select(starts_with("product"))


last_year_1 <-  cbind(last_year,  data$Proudct.issue.consequence)
last_year_2 <- cbind(last_two_year, data$Proudct.issue.consequence)
last_year_4 <- cbind(last_four_year, data$Proudct.issue.consequence)
product_1 <- cbind(product, data$Proudct.issue.consequence)
```

Descriptive Statistics of all the columns

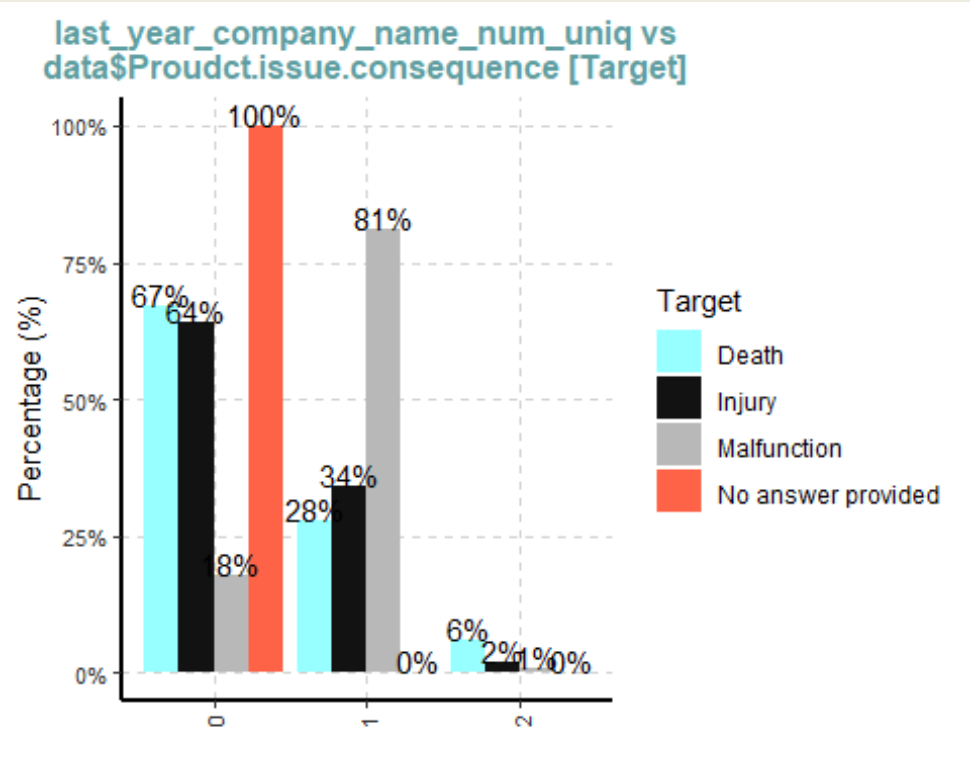
```
SmartEDA::ExpCatViz(data = last_year_1, target =
"data$Proudct.issue.consequence" )[1:3]

## [[1]]
```



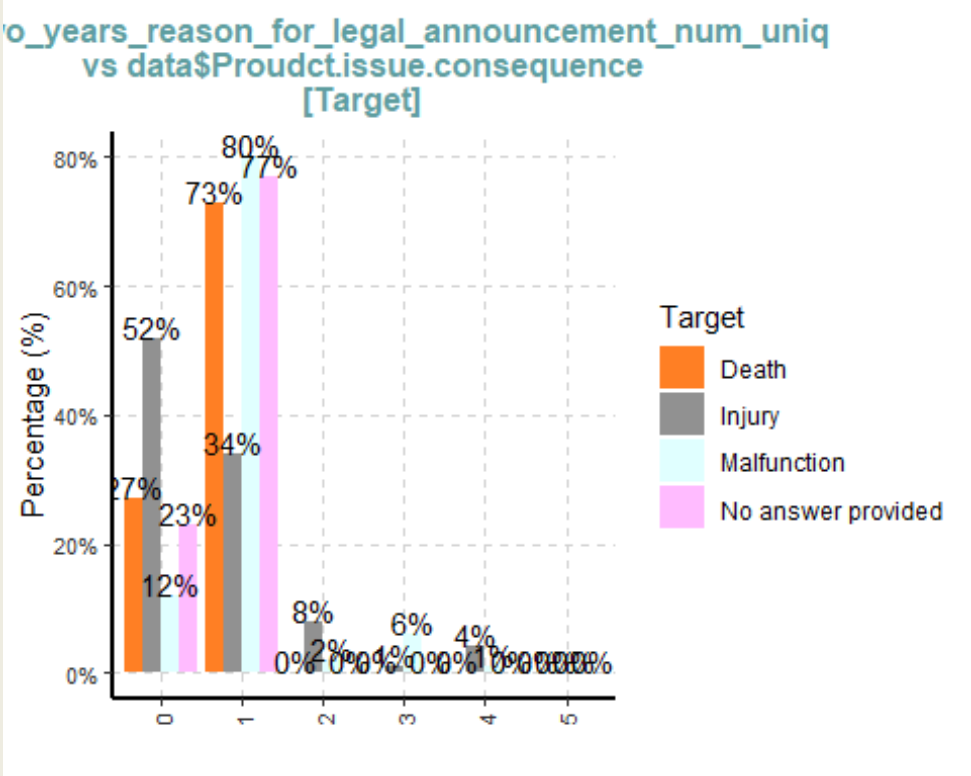

```
##
## [[2]]
```


last_year_brand_name_num_uniq vs
data$Proudct.issue.consequence [Target]
Percentage (%)
100%
75%
50%
25%
0%
0
1
2
3
4
5
6
Target
Death
Injury
Malfunction
No answer provided

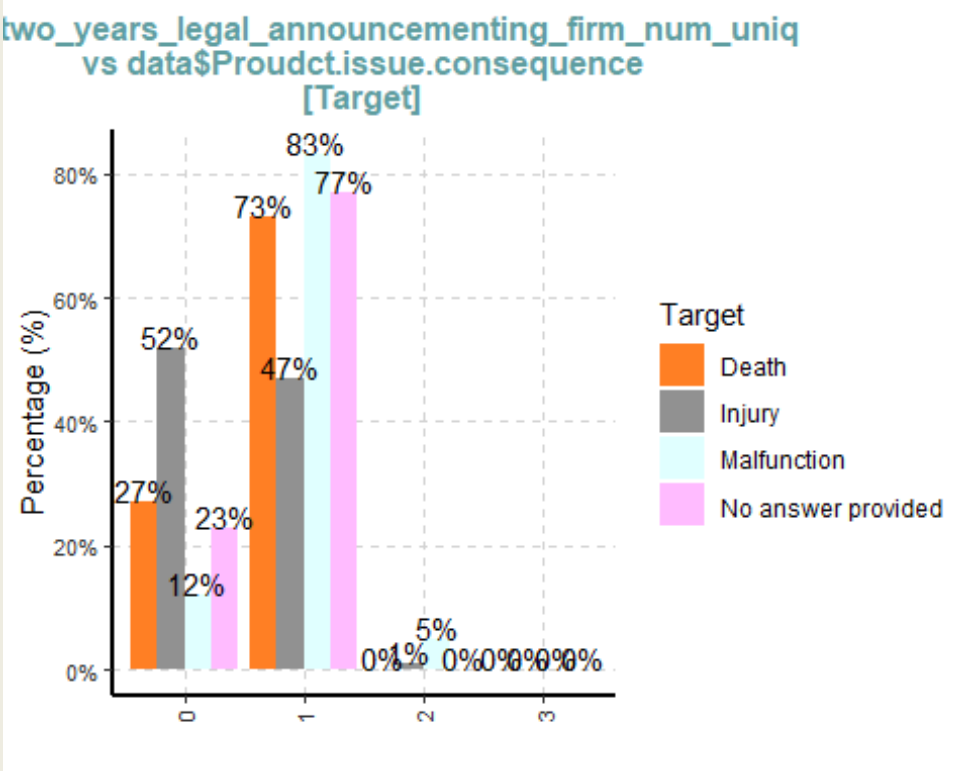
```
##
## [[3]]
```


last_year_company_name_num_uniq vs
data$Proudct.issue.consequence [Target]
Percentage (%)
100%
75%
50%
25%
0%
0
1
2
Target
Death
Injury
Malfunction
No answer provided

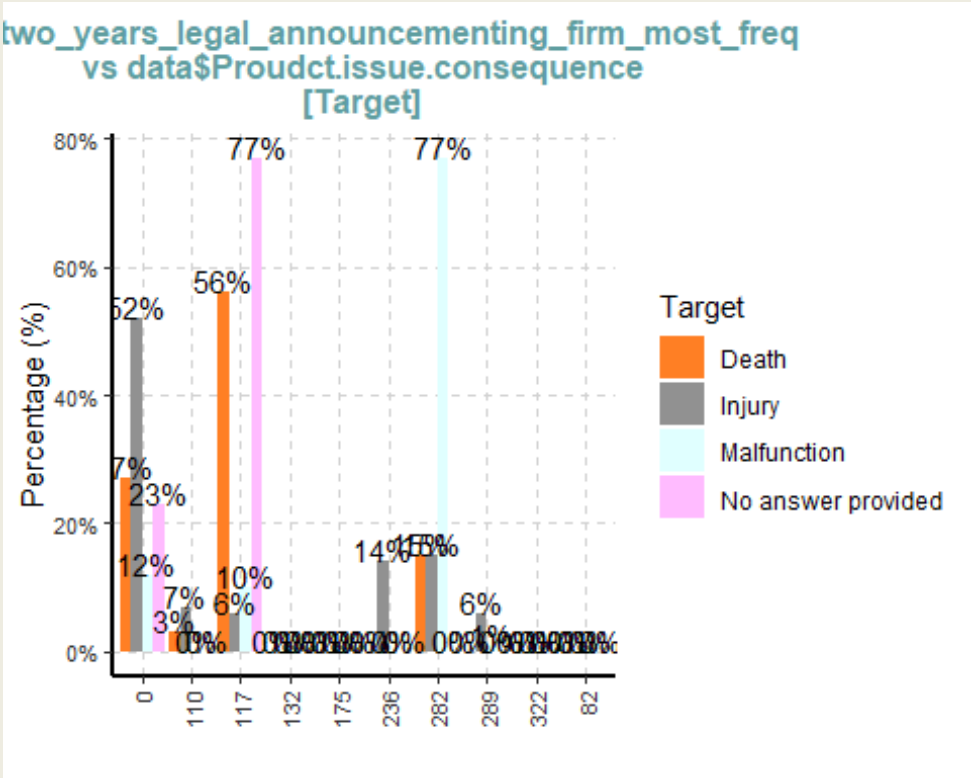
```
SmartEDA::ExpCatViz(data = last_year_2, target =
"data$Proudct.issue.consequence")[5:7]

## [[1]]
```



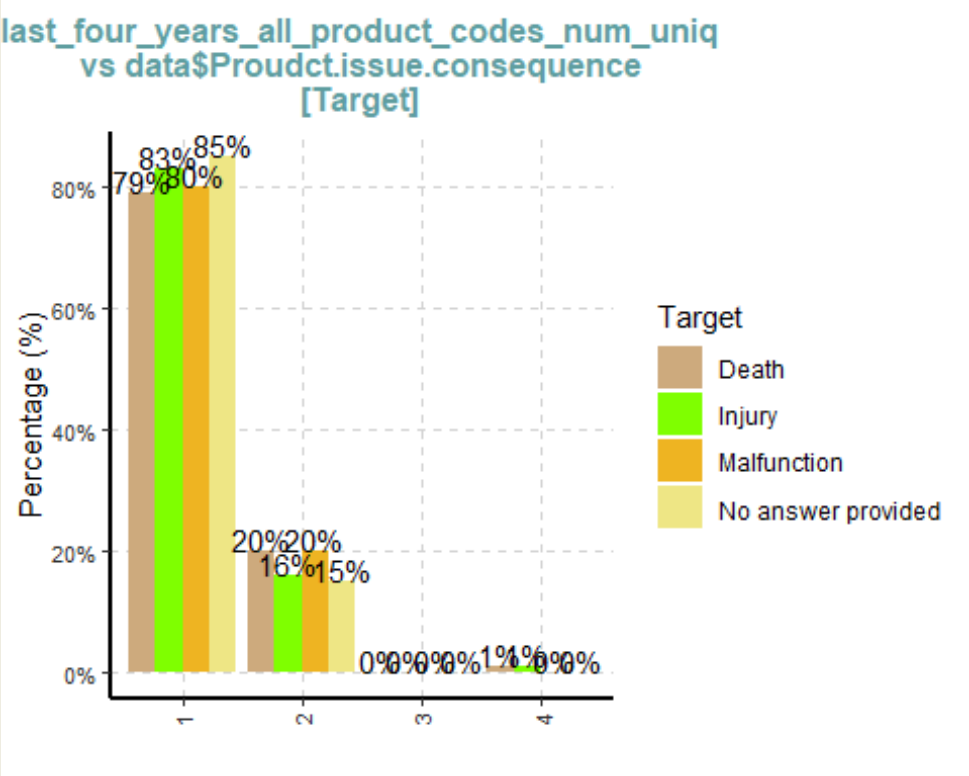

```
##
## [[2]]
```


two_years_legal_announcementing_firm_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
80%
60%
40%
20%
0%
27%
52%
12%
23%
73%
47%
83%
77%
0%
1%
5%
0%
0
1
2
3
Target
Death
Injury
Malfunction
No answer provided

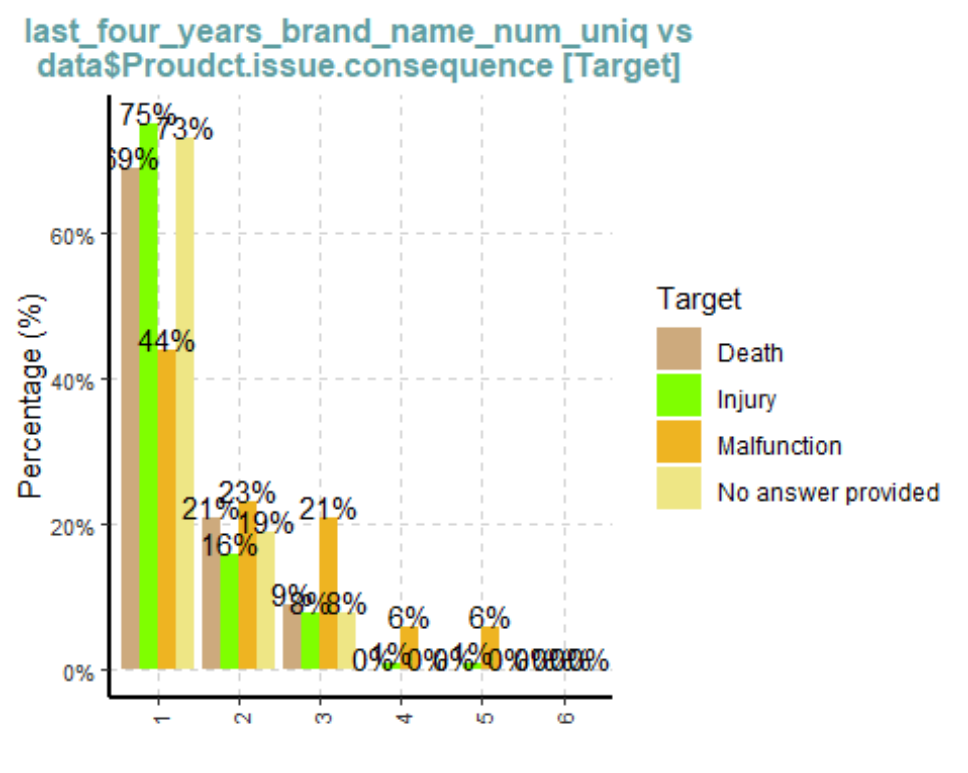
```
##
## [[3]]
```


two_years_legal_announcementing_firm_most_freq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
80%
60%
40%
20%
0%
77%
77%
56%
52%
23%
12%
10%
7%
6%
3%
14%
6%
0
110
117
132
175
236
282
289
322
82
Target
Death
Injury
Malfunction
No answer provided

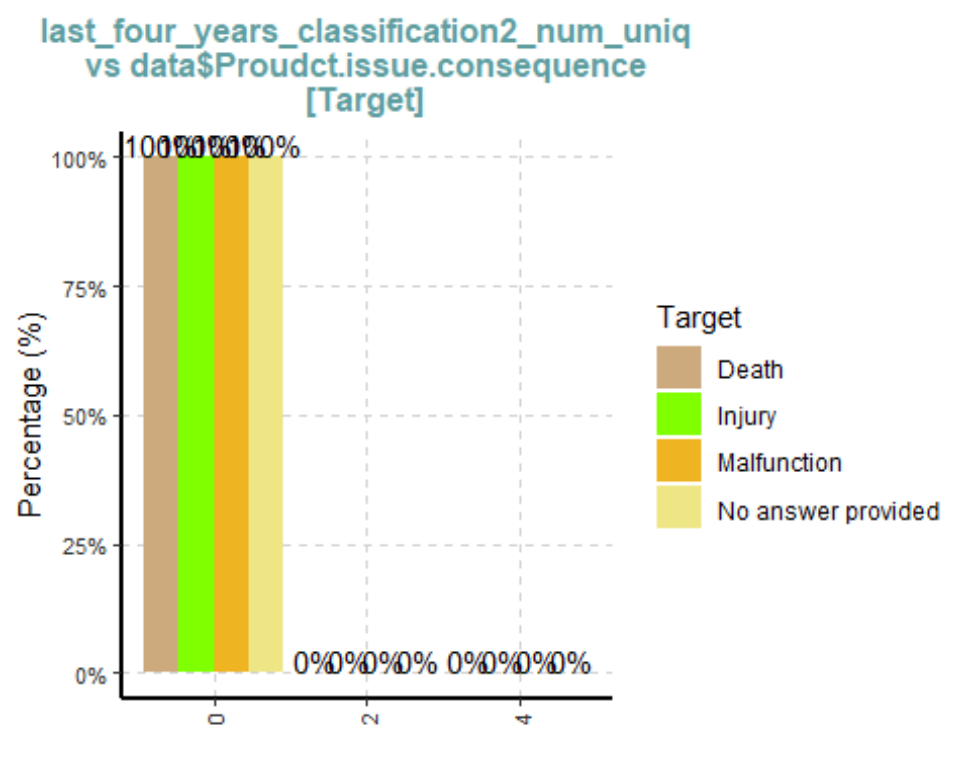
```
SmartEDA::ExpCatViz(data = last_year_4, target =
"data$Proudct.issue.consequence")[1:3]

## [[1]]
```



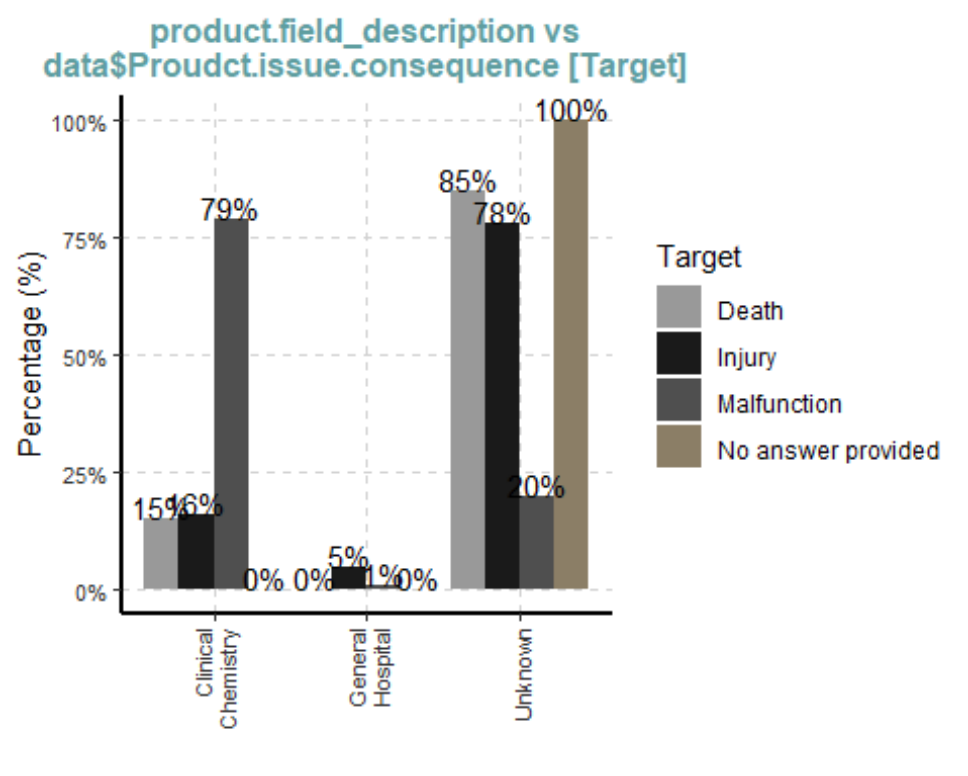

```
##
## [[2]]
```


last_four_years_brand_name_num_uniq vs data$Proudct.issue.consequence [Target]
Percentage (%)
60%
40%
20%
0%
75%
73%
44%
23%
21%
19%
16%
21%
9%
8%
8%
6%
6%
1%
1%
0%
1
2
3
4
5
6
Target
Death
Injury
Malfunction
No answer provided

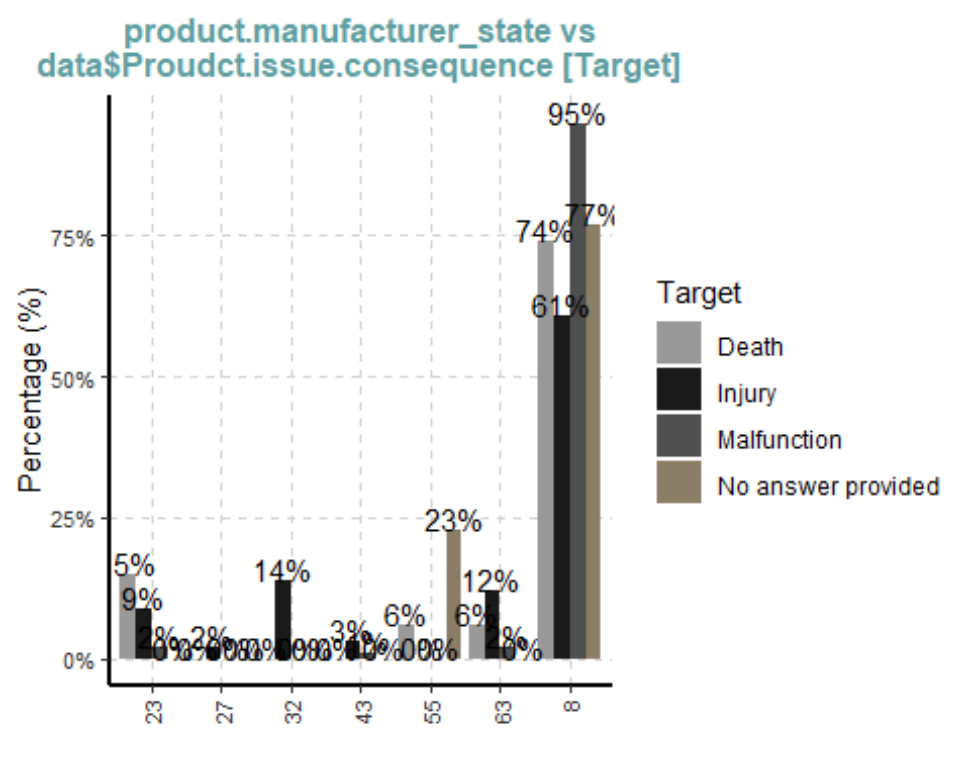
```
##
## [[3]]
```


last_four_years_classification2_num_uniq vs data$Proudct.issue.consequence [Target]
Percentage (%)
100%
75%
50%
25%
0%
0%
0
2
4
Target
Death
Injury
Malfunction
No answer provided

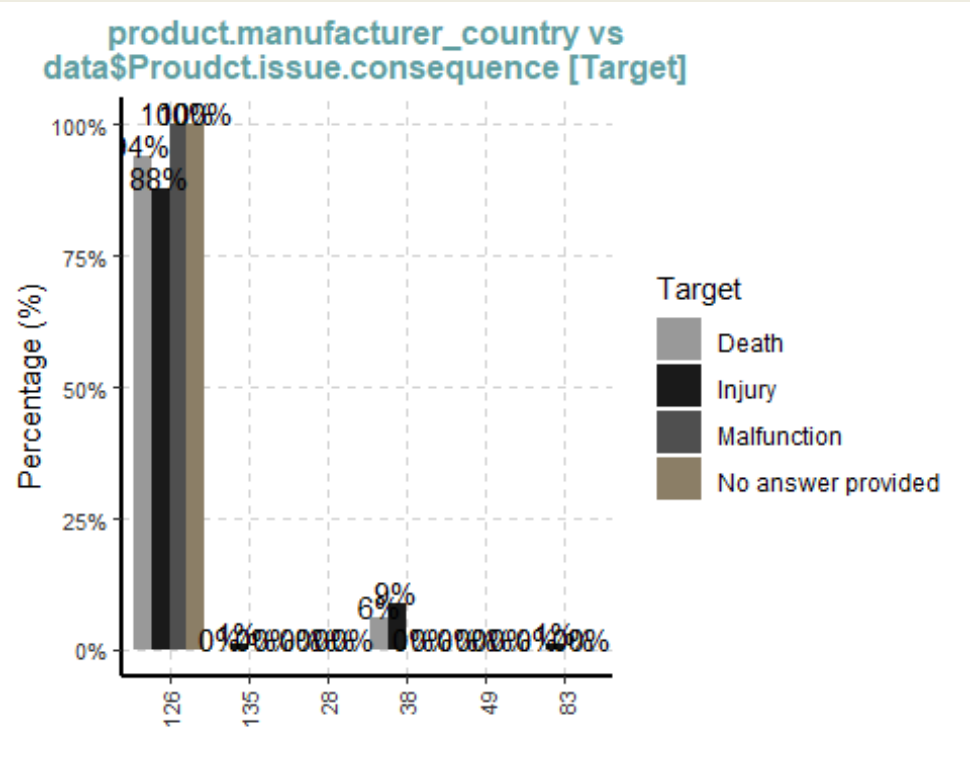
```
SmartEDA::ExpCatViz(data = product_1, target =
"data$Proudct.issue.consequence")

## [[1]]
```




```
##
## [[2]]
```


product.manufacturer_state vs
data$Proudct.issue.consequence [Target]
Percentage (%)
75%
50%
25%
0%
95%
74%
77%
61%
23%
14%
12%
6%
6%
5%
9%
3%
23
27
32
43
55
63
8
Target
Death
Injury
Malfunction
No answer provided

```
##
## [[3]]
```


product.manufacturer_country vs
data$Proudct.issue.consequence [Target]
Percentage (%)
100%
75%
50%
25%
0%
88%
9%
6%
126
135
28
38
49
83
Target
Death
Injury
Malfunction
No answer provided

POSSIBLE INSIGHTS FROM THE GIVEN DATASET

Exploring the data from the given dataset reveals several key insights that can inform our understanding of product issues and their consequences, like death, injury, malfunction, or no specified outcome. These insights have implications for quality control, risk management, and regulatory compliance in product manufacturing and distribution.

Here are a few key observations:

Product Issues and Consequences Relationship: The data visualizations suggest a strong relationship between unique product codes, manufacturer states, and countries with the frequency of product issues like injury, malfunction, and more severely, death. These insights highlight the importance of traceability in the product supply chain, as certain product batches or locations are more frequently associated with adverse outcomes.The 'product.manufacturer_state' and 'product.manufacturer_country' data slices provide a geographic perspective, pointing out that certain locations may have higher reports of product issues. This could reflect regional manufacturing practices, distribution patterns, or market penetration levels that influence the reported data. For example, if a particular state or country is overrepresented in the dataset for adverse product issues, it could warrant a deeper examination of local manufacturing standards, consumer usage, or reporting practices.

Geographical Impact: The visualizations indicate significant regional disparities in product issue consequences, implying that manufacturer location is a potential indicator of product safety. This could be due to local manufacturing practices, the availability of quality materials, or varying enforcement of safety standards. In turn, this might necessitate targeted interventions in specific areas to mitigate risks.In examining the relationship between 'product.manufacturer_state' and the outcomes, the data depicted a notable variation in issue frequency by geographic location, hinting at potential regional differences in manufacturing practices or regulatory compliance. The selection of the 'product.manufacturer_country' field aimed to extend this geographic analysis to a global scale, thereby enabling a comparison of domestic and international safety trends.

Reasons for Legal Announcements: The frequency of legal announcements related to product issues suggests that certain causes for these announcements are more prevalent, which might inform the prioritization of regulatory oversight. For instance, a high number of legal announcements in a specific area could indicate systemic problems that require attention from manufacturers and regulatory bodies.Considering the 'last_two_years_reason_for_legal_announcement_num_uniq' variable, it seems that certain legal announcements are more frequently associated with product issues

Root Cause Analysis: The data points to certain root causes being more prevalent in product failures leading to legal consequences. Understanding these root causes is key to preventing future issues and improving overall product safety. Manufacturers might need to investigate these root causes more deeply and develop corrective actions to address them.For example.The variables such as 'last_year_all_product_codes_num_uniq' and 'last_year_brand_name_num_uniq' show varying degrees of association with different consequences, suggesting a relationship between the diversity of products or brands and

the incidence of issues. It can be inferred that a wider range of product codes or brands could be correlated with an increased likelihood of product issues being reported. Moreover, this could indicate that certain product types or brands may be more prone to issues, or conversely, that they are simply more frequently used and hence more often implicated in reports.

Product Field Descriptions: Different product field descriptions show varying levels of association with product issues. Fields like 'Clinical Chemistry' might be particularly sensitive due to their direct impact on health. This indicates the need for a more stringent regulatory framework and continuous monitoring in these fields to ensure public safety.

Predictive Insights: By analyzing the frequency and type of product issues in relation to the product descriptions, manufacturers and stakeholders can identify patterns that may be predictive of future issues. This could enable proactive measures to prevent issues rather than reacting to them after they occur.

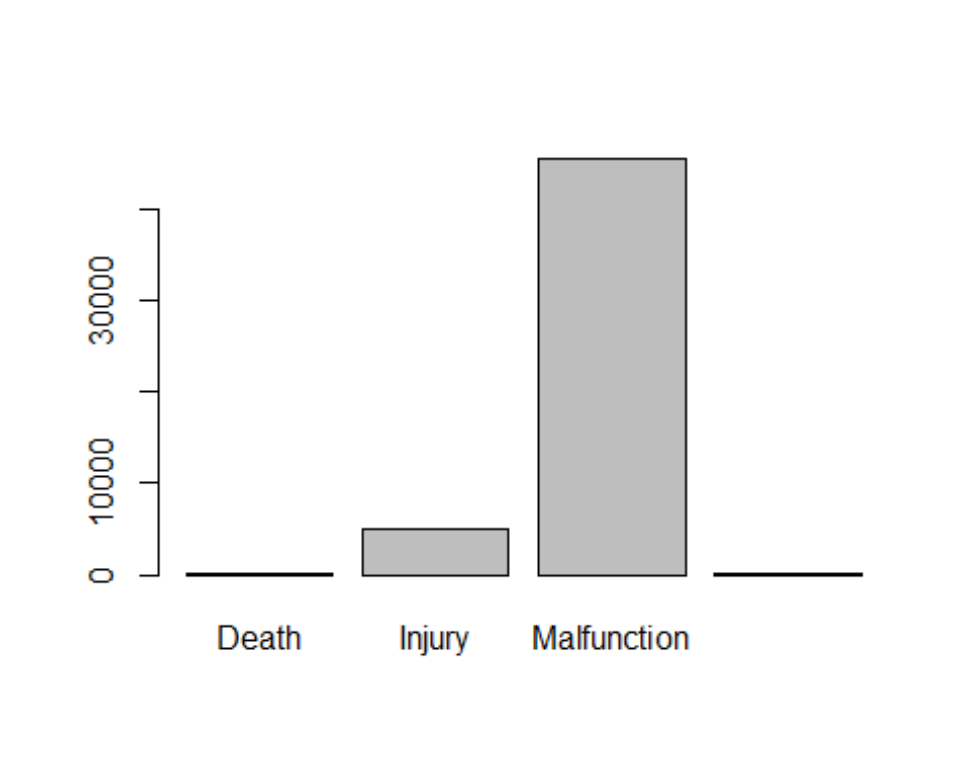
Global Comparisons: The manufacturer country data provides a global perspective, suggesting that international standards might not be uniformly applied or enforced. Collaboration on a global scale might be required to ensure product safety across borders.

In summary, these data visualizations underscore the interconnectivity between product safety issues, geographical manufacturing locations, and legal consequences.Moreover, these insights can drive policy changes and lead to the development of best practices that ensure consumer safety and maintain public trust in products, especially in high-stake fields like 'clinical-chemical' products.

STATISTICAL LEARNING MODEL BUILDING

1) Barplot before defining the levels of target variable

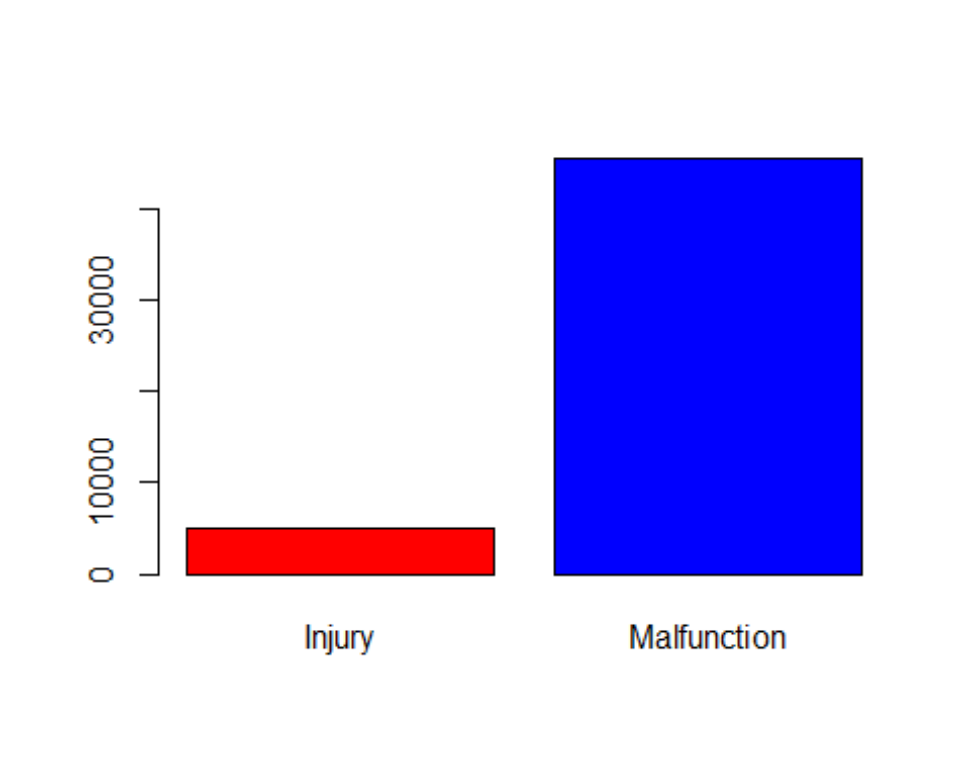
```
target_before_def <- table(data$Proudct.issue.consequence)

barplot(target_before_def)
```




```
data <- data %>% mutate(Issue_Consequence_new = case_when(
  Proudct.issue.consequence == "Death" ~ "Malfunction",
   Proudct.issue.consequence == "Other" ~ "Malfunction",
   Proudct.issue.consequence == "No answer provided" ~ "Malfunction",
  Proudct.issue.consequence == "Malfunction" ~ "Malfunction",
  TRUE ~ "Injury")
  )


data <- data %>% dplyr::select(-c(Proudct.issue.consequence))
```

#Barplot after defining the levels of target variable

```
target_after_def <- table(data$Issue_Consequence_new)

barplot(target_after_def, col = c("red","blue"))
```



Lasso regression, In this phase of the assignment, the task is to select one product.field_description from the dataset to investigate the impact of various predictors on outcomes. To manage the complexity and potential overfitting associated with a large dataset, the strategy involves dividing the dataset into manageable subsets for detailed analysis. In this context, the dataset has been segmented into three subsets: subset_1 comprises the first 16,000 records, subset_2 includes records from 16,001 to 32,000, and subset_3 encompasses records from 32,001 to 50,646. This division facilitates a focused approach to running Lasso regression models on each subset, allowing for the identification and comparison of critical predictors across different segments of the dataset.

The use of Lasso regression is particularly apt for this task due to its ability to perform both variable selection and regularization, thereby enhancing the model's predictive accuracy and interpretability by eliminating non-informative predictors. By applying Lasso regression to each subset individually, students can uncover the top critical predictors within varying contexts of product.field_description, ensuring a comprehensive analysis that accounts for the dataset's inherent diversity. This methodological choice not only mitigates the risk of overfitting but also addresses the analytical challenges posed by datasets with a high dimensionality of predictors. The insights derived from these models will be instrumental in understanding the factors that significantly influence product outcomes, guiding future product development and quality control strategies.

Generation of Subsets

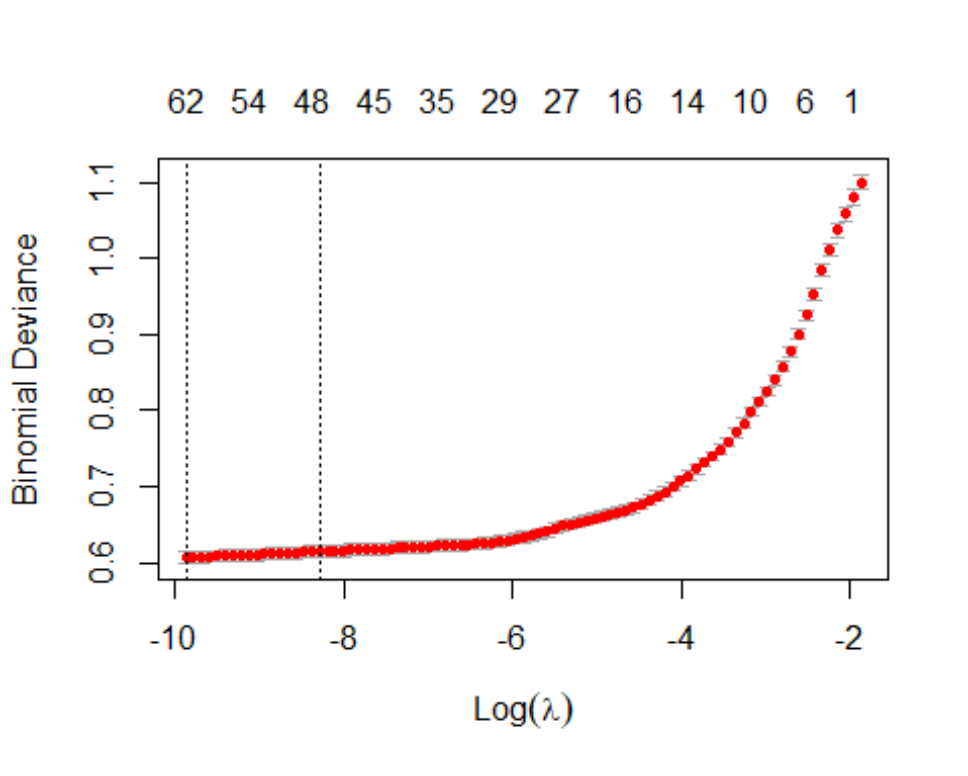
```
subset_1 <- data[1: 16000,]
subset_2 <- data[16001:32000,]
subset_3 <- data[32001:50646,]
```

#Created the train-test-split for the subsets

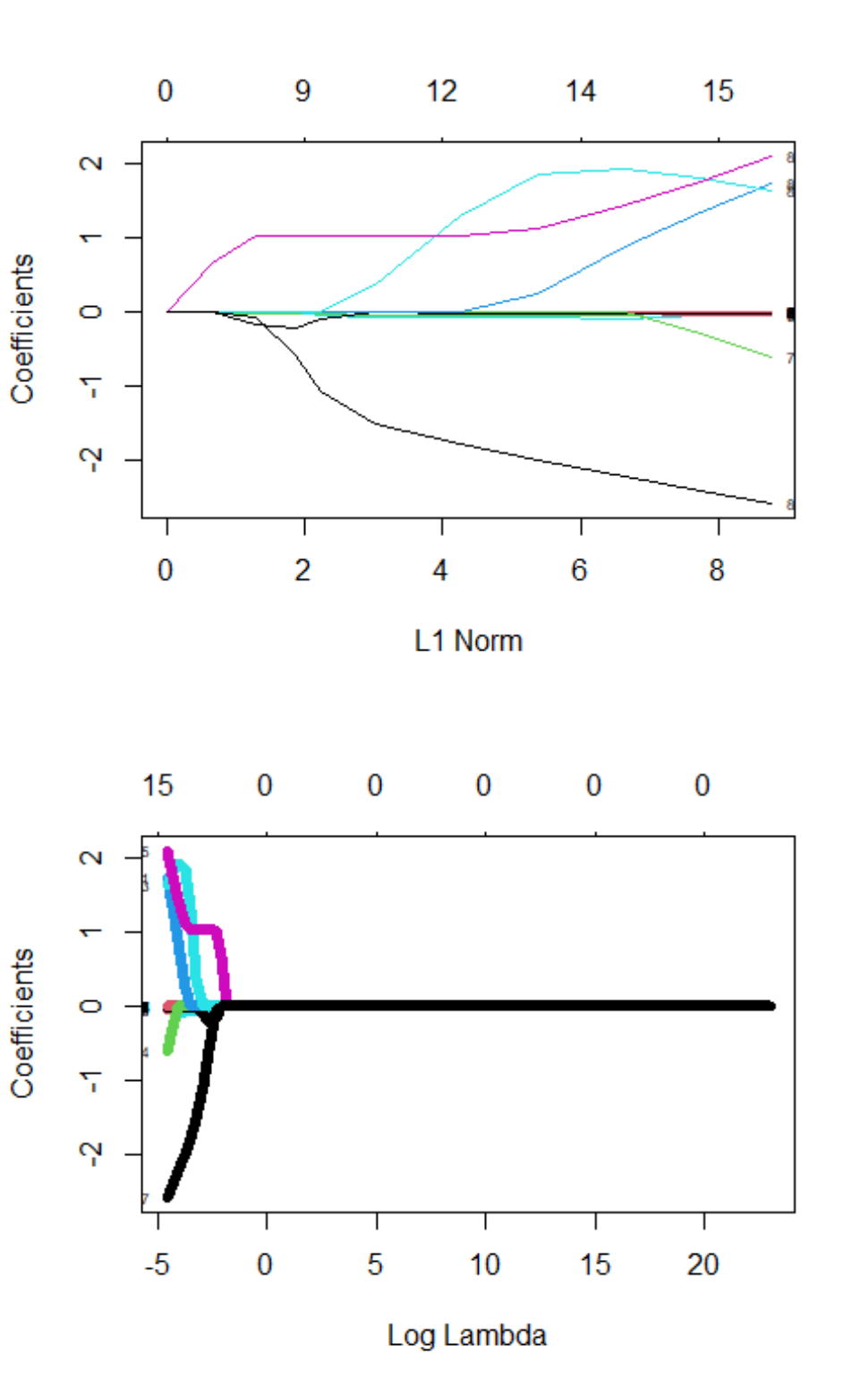
```
subset1_sets <- train_test_split(subset_1)
subset2_sets <- train_test_split(subset_2)
subset3_sets <- train_test_split(subset_3)
```

#LASSO REGRESSION for subset1

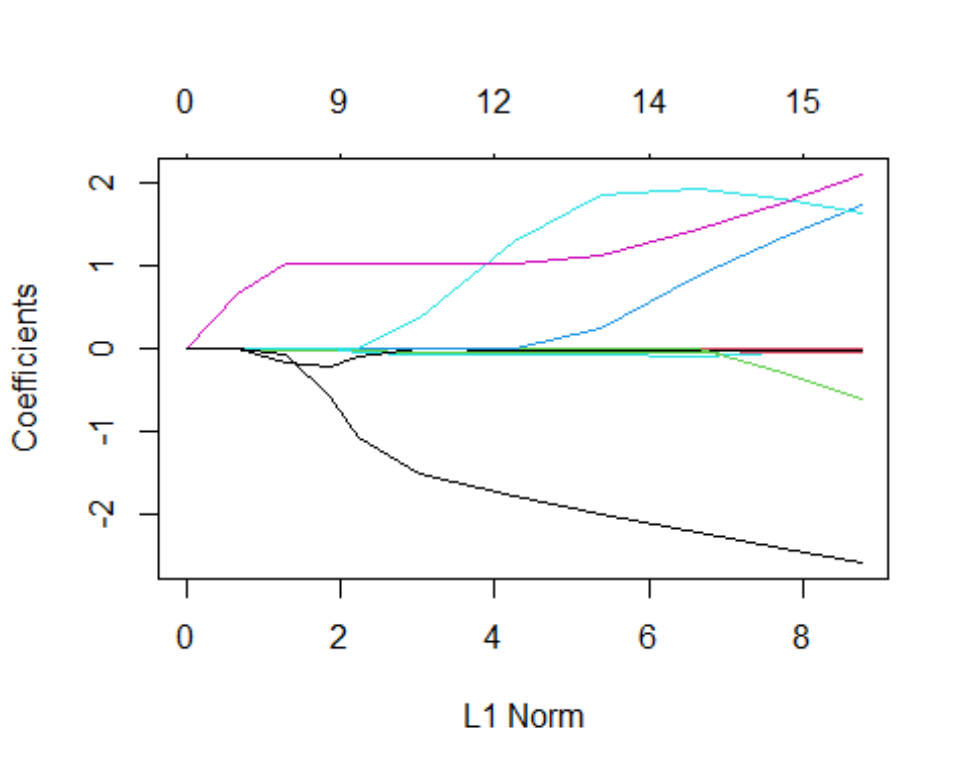
```
lasso_subset1 <- lasso(x_train = subset1_sets$X_train, y_train =
subset1_sets$y_train, x_test = subset1_sets$x_test,
                       y_test =subset1_sets$y_test,subsets = subset_1)
```




```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```




```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



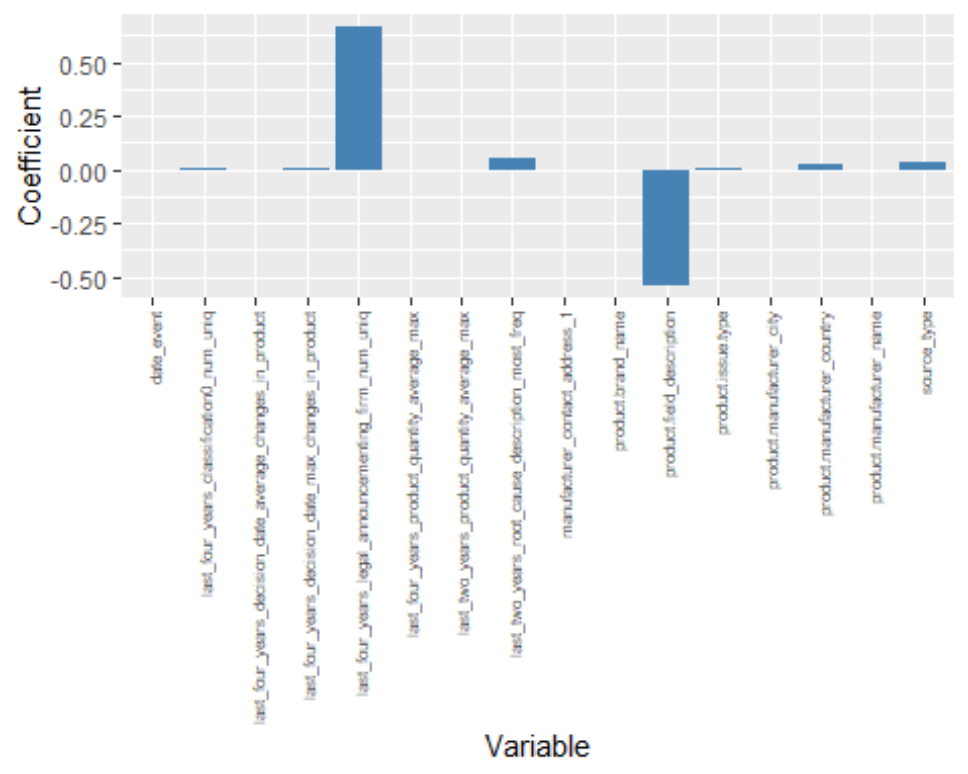

```
##
variable
## date_event
date_event
## last_two_years_root_cause_description_most_freq
last_two_years_root_cause_description_most_freq
## last_two_years_product_quantity_average_max
last_two_years_product_quantity_average_max
## last_four_years_classification0_num_uniq
last_four_years_classification0_num_uniq
## last_four_years_legal_announcementing_firm_num_uniq
last_four_years_legal_announcementing_firm_num_uniq
## last_four_years_product_quantity_average_max
last_four_years_product_quantity_average_max
## last_four_years_decision_date_max_changes_in_product
last_four_years_decision_date_max_changes_in_product
## last_four_years_decision_date_average_changes_in_product
last_four_years_decision_date_average_changes_in_product
## manufacturer_contact_address_1
manufacturer_contact_address_1
## product.brand_name
product.brand_name
## product.issue.type
product.issue.type
## source_type
source_type
## product.manufacturer_name
```

```
product.manufacturer_name
## product.manufacturer_city
product.manufacturer_city
## product.manufacturer_country
product.manufacturer_country
## product.field_description
product.field_description
##                                                        coefficient
## date_event                                          -1.143300e-03
## last_two_years_root_cause_description_most_freq      5.147681e-02
## last_two_years_product_quantity_average_max         -7.428853e-07
## last_four_years_classification0_num_uniq             1.032031e-03
## last_four_years_legal_announcementing_firm_num_uniq  6.643917e-01
## last_four_years_product_quantity_average_max        -7.752786e-08
## last_four_years_decision_date_max_changes_in_product 8.566777e-03
## last_four_years_decision_date_average_changes_in_product 5.063915e-16
## manufacturer_contact_address_1                       1.664532e-04
## product.brand_name                                   2.896690e-06
## product.issue.type                                   4.024719e-03
## source_type                                          3.154337e-02
## product.manufacturer_name                           -1.552065e-05
## product.manufacturer_city                            1.413947e-04
## product.manufacturer_country                         2.211264e-02
## product.field_description                           -5.384179e-01
```

#creating a summary plot showing the differences between sets of top 10 critical predictors identified across fields

```
lasso_subset1$top10

##
variable
## last_four_years_legal_announcementing_firm_num_uniq
last_four_years_legal_announcementing_firm_num_uniq
## product.field_description
product.field_description
## last_two_years_root_cause_description_most_freq
last_two_years_root_cause_description_most_freq
## source_type
source_type
## product.manufacturer_country
product.manufacturer_country
## last_four_years_decision_date_max_changes_in_product
last_four_years_decision_date_max_changes_in_product
## product.issue.type
product.issue.type
## date_event
date_event
## last_four_years_classification0_num_uniq
last_four_years_classification0_num_uniq
```

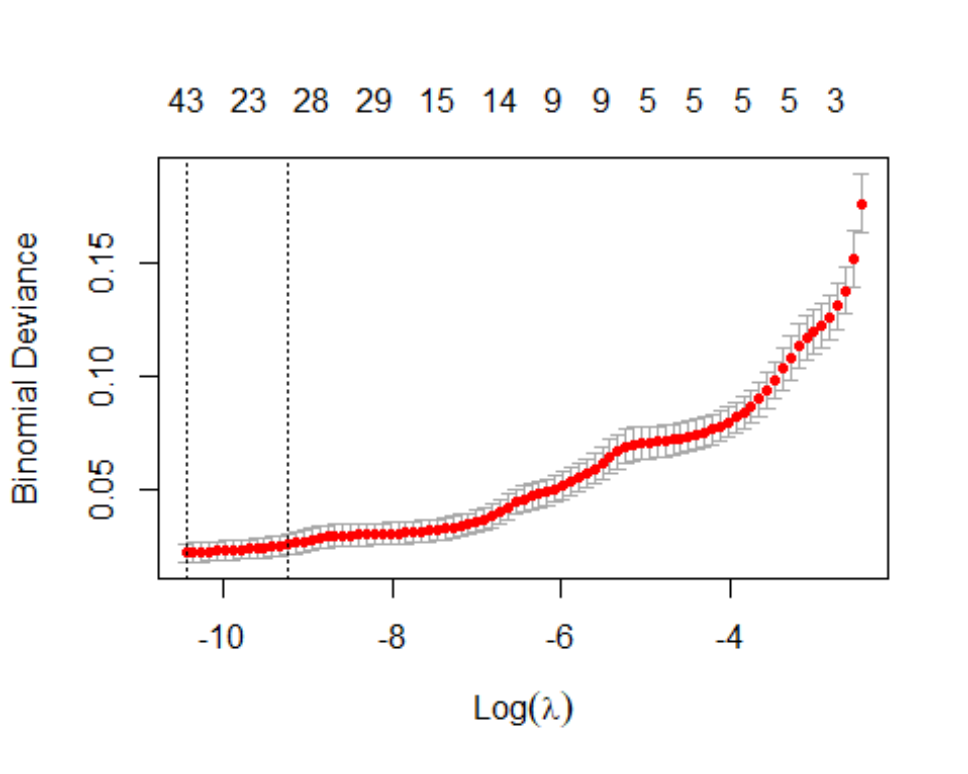
```
## manufacturer_contact_address_1
manufacturer_contact_address_1
##                                                          coefficient
## last_four_years_legal_announcementing_firm_num_uniq    0.6643917361
## product.field_description                             -0.5384179334
## last_two_years_root_cause_description_most_freq        0.0514768052
## source_type                                            0.0315433706
## product.manufacturer_country                           0.0221126422
## last_four_years_decision_date_max_changes_in_product   0.0085667771
## product.issue.type                                     0.0040247195
## date_event                                            -0.0011432996
## last_four_years_classification0_num_uniq               0.0010320312
## manufacturer_contact_address_1                         0.0001664532

lasso_subset1$plot_top_10
```



#LASSO REGRESSION for subset2

```
lasso_subset2 <- lasso(x_train = subset2_sets$X_train, y_train =
subset2_sets$y_train, x_test = subset2_sets$x_test,
                       y_test =subset2_sets$y_test,subsets = subset_2)

## Warning: from glmnet C++ code (error code -93); Convergence for 93th
lambda
## value not reached after maxit=100000 iterations; solutions for larger
lambdas
## returned
```

```
## Warning in plotCoef(x$beta, lambda = x$lambda, df = x$df, dev =
x$dev.ratio, :
## No plot produced since all coefficients zero

## Warning in plotCoef(x$beta, lambda = x$lambda, df = x$df, dev =
x$dev.ratio, :
## No plot produced since all coefficients zero

## Warning in plotCoef(x$beta, lambda = x$lambda, df = x$df, dev =
x$dev.ratio, :
## No plot produced since all coefficients zero

## Warning: from glmnet C++ code (error code -93); Convergence for 93th
lambda
## value not reached after maxit=100000 iterations; solutions for larger
lambdas
## returned
```




```
## [1] variable    coefficient
## <0 rows> (or 0-length row.names)
```

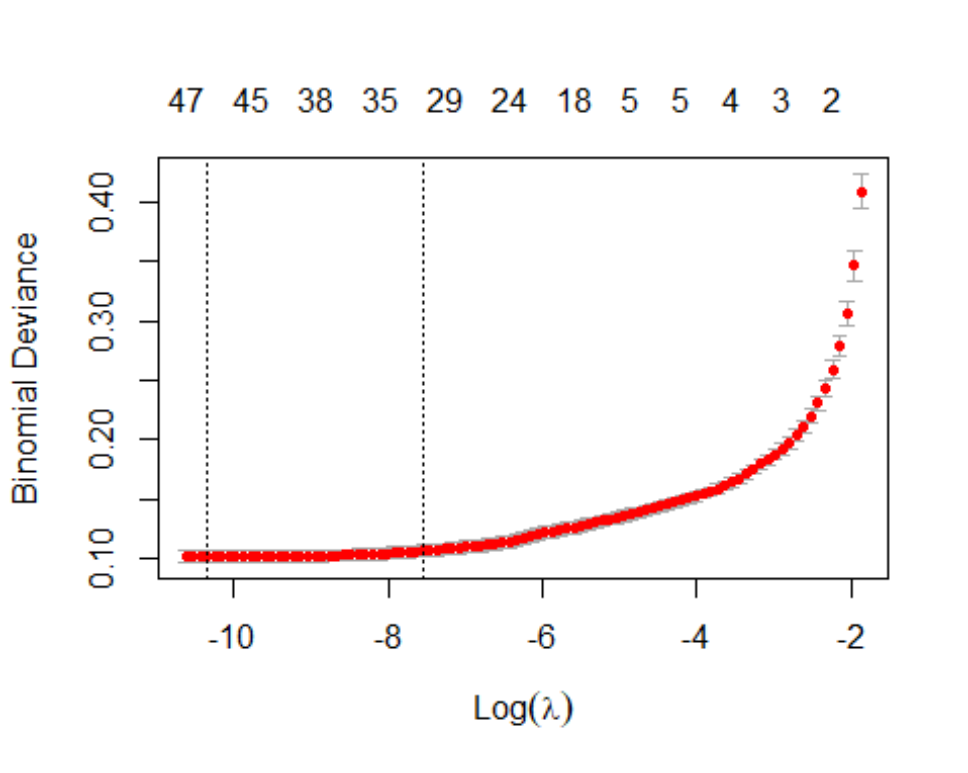
#creating a summary plot showing the differences between sets of critical predictors identified across fields

```
lasso_subset2$top10
```

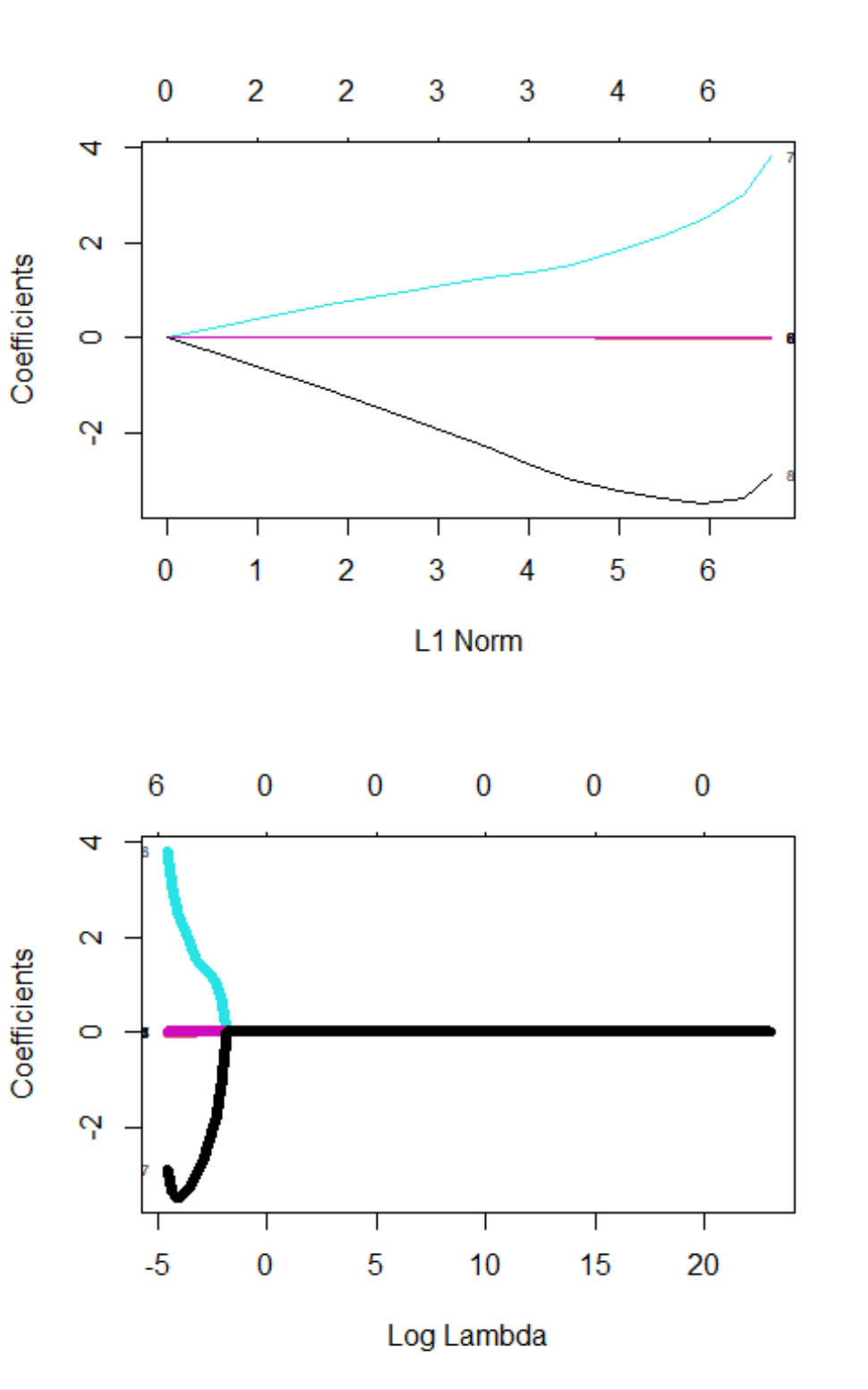
```
## [1] variable    coefficient
## <0 rows> (or 0-length row.names)

lasso_subset2$plot_top_10
```

#LASSO REGRESSION for subset3

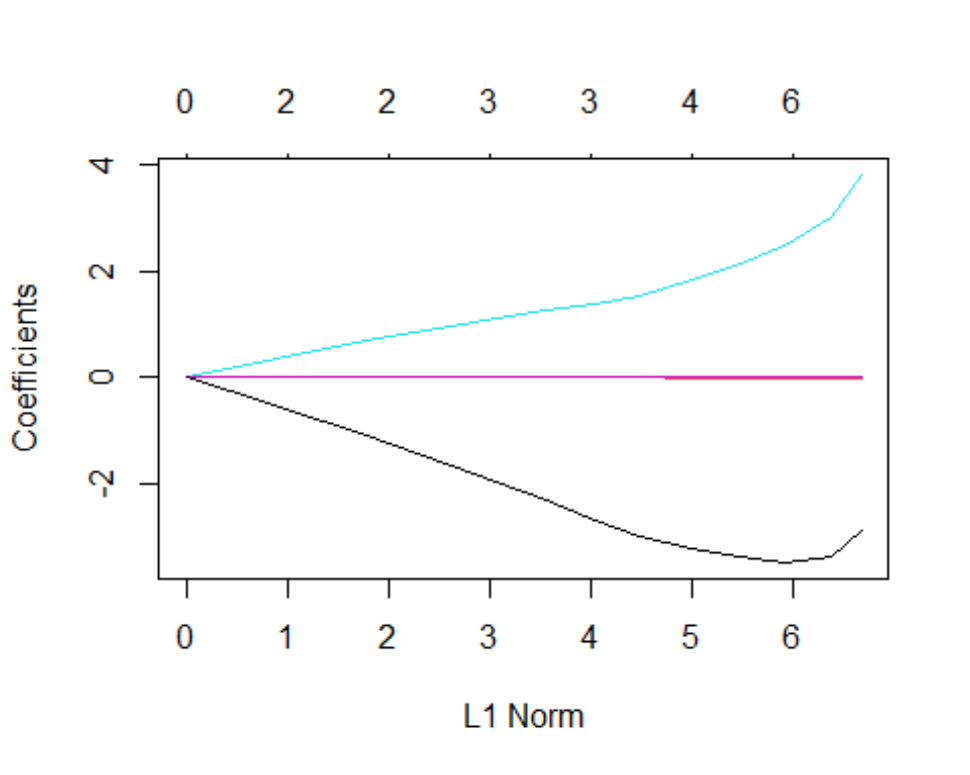
```
lasso_subset3 <- lasso(x_train = subset3_sets$X_train, y_train =
subset3_sets$y_train, x_test = subset3_sets$x_test,
                       y_test =subset3_sets$y_test,subsets = subset_3)
```




```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



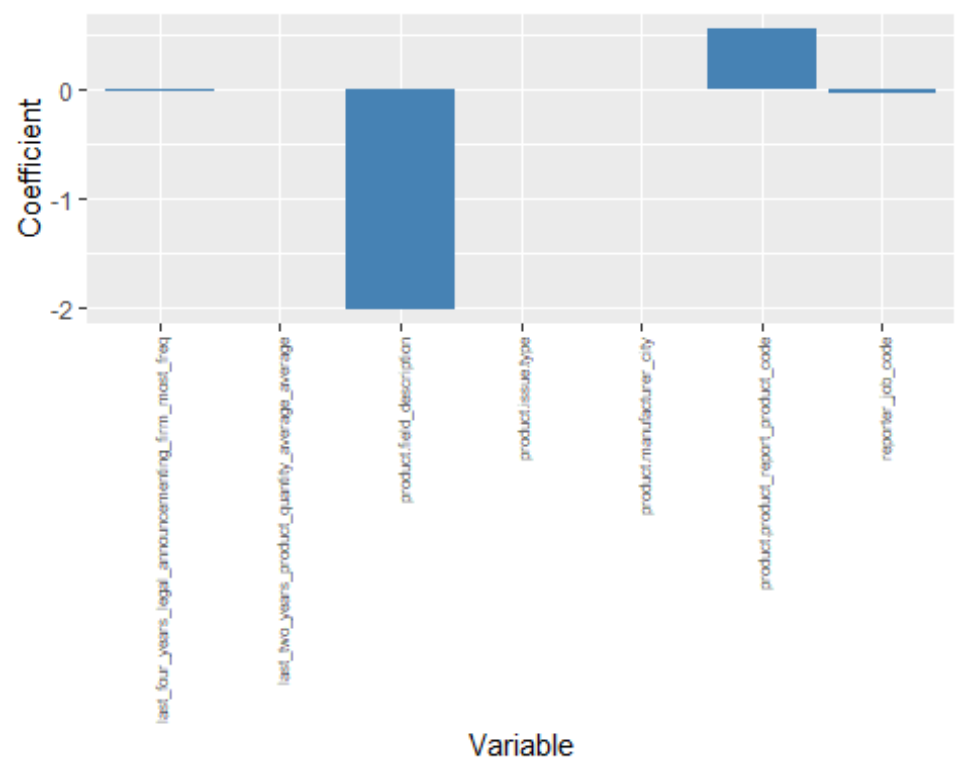

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```




```
##
variable
## last_two_years_product_quantity_average_average
last_two_years_product_quantity_average_average
## last_four_years_legal_announcementing_firm_most_freq
last_four_years_legal_announcementing_firm_most_freq
## product.issue.type
product.issue.type
## reporter_job_code
reporter_job_code
## product.manufacturer_city
product.manufacturer_city
## product.field_description
product.field_description
## product.product_report_product_code
product.product_report_product_code
##                                                    coefficient
## last_two_years_product_quantity_average_average      -7.413879e-07
## last_four_years_legal_announcementing_firm_most_freq -7.543105e-03
## product.issue.type                                    6.694810e-03
## reporter_job_code                                    -3.776896e-02
## product.manufacturer_city                             8.571194e-05
## product.field_description                            -2.027715e+00
## product.product_report_product_code                   5.695274e-01
```

#creating a summary plot showing the differences between sets of critical predictors identified across fields

```
lasso_subset3$top10

##
variable
## product.field_description
product.field_description
## product.product_report_product_code
product.product_report_product_code
## reporter_job_code
reporter_job_code
## last_four_years_legal_announcementing_firm_most_freq
last_four_years_legal_announcementing_firm_most_freq
## product.issue.type
product.issue.type
## product.manufacturer_city
product.manufacturer_city
## last_two_years_product_quantity_average_average
last_two_years_product_quantity_average_average
##                                                    coefficient
## product.field_description                         -2.027715e+00
## product.product_report_product_code                5.695274e-01
## reporter_job_code                                 -3.776896e-02
## last_four_years_legal_announcementing_firm_most_freq -7.543105e-03
## product.issue.type                                 6.694810e-03
## product.manufacturer_city                          8.571194e-05
## last_two_years_product_quantity_average_average   -7.413879e-07

lasso_subset3$plot_top_10
```



#Comparing lasso for all the three subsets

```
lasso_subset1$accuracy
```

```
## [1] 0.8025
```

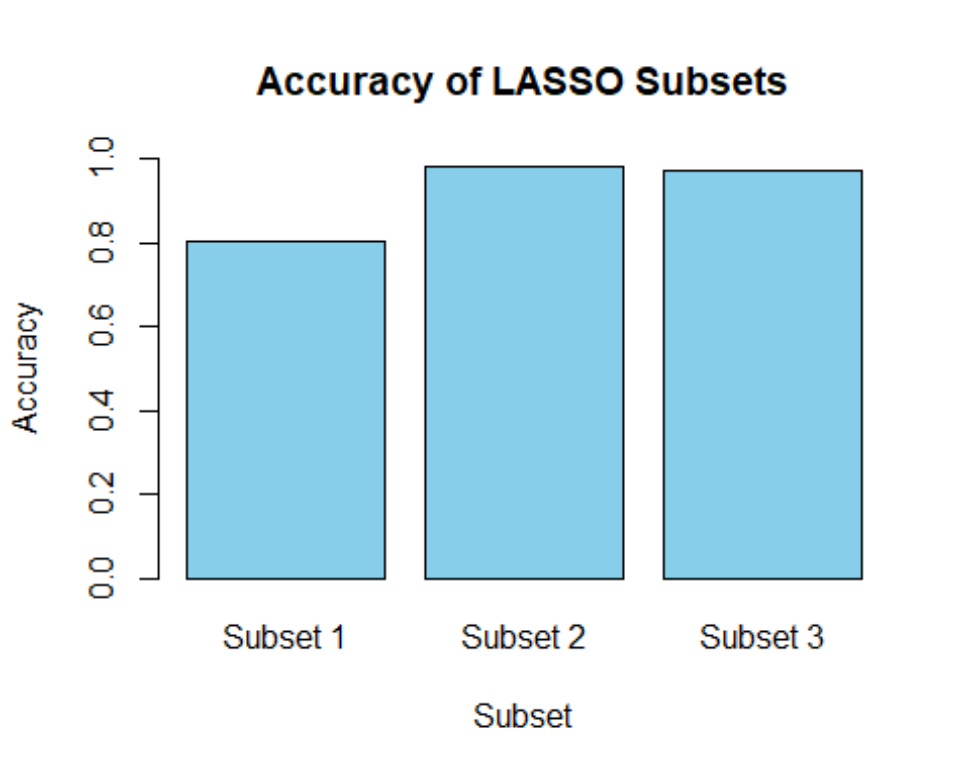
```
lasso_subset2$accuracy
```

```
## [1] 0.9829167
```

```
lasso_subset3$accuracy
```

```
## [1] 0.9730068
```

```
# Assuming lasso_subset1, lasso_subset2, lasso_subset3 have an 'accuracy'
numeric field
accuracy_values <- c(lasso_subset1$accuracy, lasso_subset2$accuracy,
lasso_subset3$accuracy)
names(accuracy_values) <- c("Subset 1", "Subset 2", "Subset 3")

# Creating the bar plot
barplot(accuracy_values,
        main="Accuracy of LASSO Subsets",
        xlab="Subset",
        ylab="Accuracy",
        col="skyblue",
        ylim=c(0, 1))  # Assuming accuracy is a proportion between 0 and 1
```



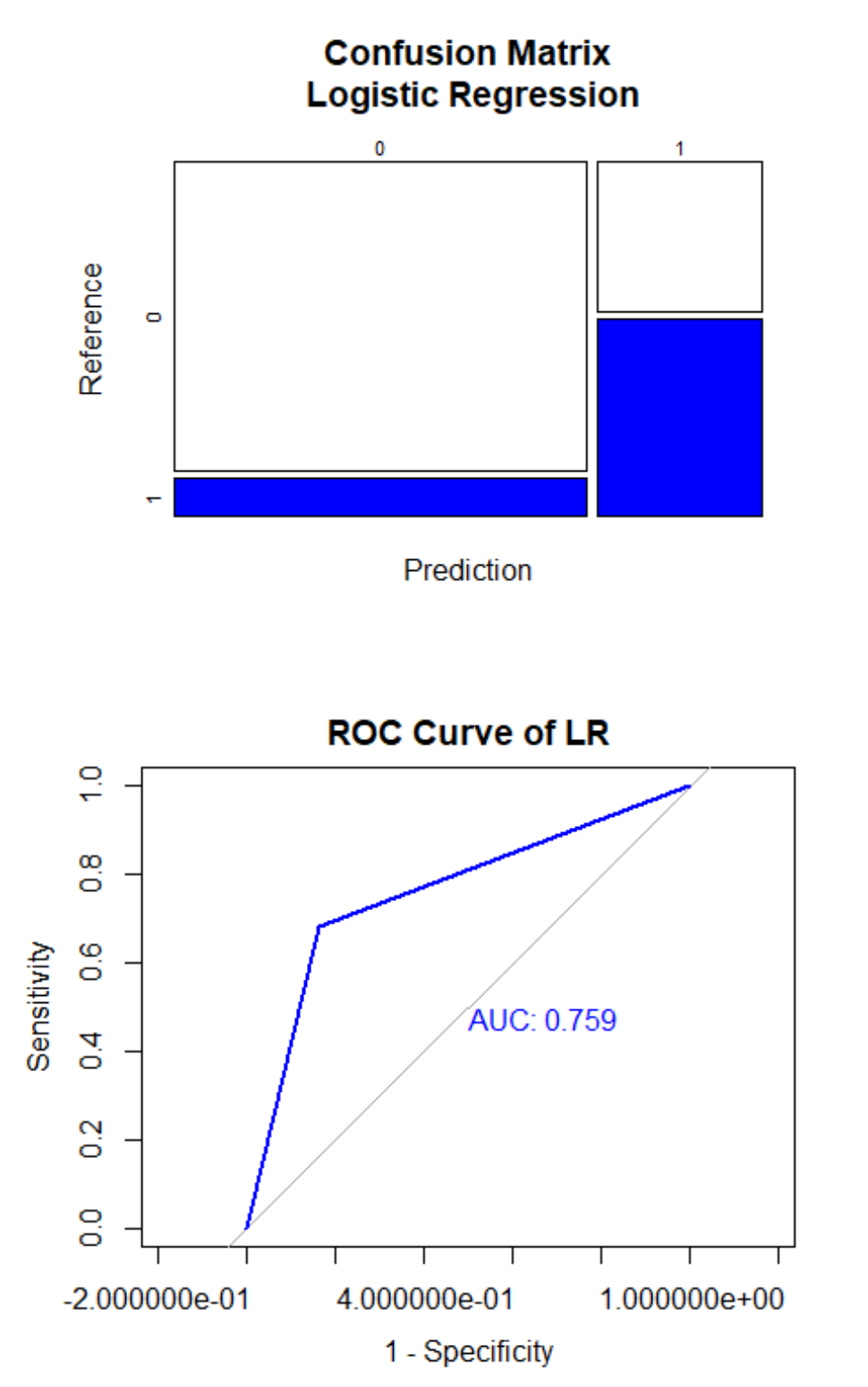
#FINDINGS -LASSO REGRESSION

The Lasso regression analysis conducted on three distinct subsets of the dataset reveals significant insights into the critical predictors influencing the outcomes within the context of product safety and quality. For subset 1, the top predictors include variables related to legal announcements, product field descriptions, root cause descriptions, and the product's country of manufacture, indicating a complex interplay between regulatory actions, product characteristics, and geographical origins in determining product outcomes. Notably, the last_four_years_legal_announcementing_firm_num_uniq variable emerged as a significant positive predictor, suggesting that firms with a higher number of unique legal announcements in the past four years are associated with increased product issues, possibly reflecting ongoing compliance or quality challenges within these firms. In contrast, the analysis for subset 2 encountered convergence issues, suggesting that the predictors in this segment of the dataset might not significantly influence the outcomes or that the model's complexity exceeded the available data's capacity to provide meaningful insights. This indicates a potential need for further data preprocessing or model adjustment to achieve convergence and extract valuable insights. Subset 3's analysis, however, identified a different set of critical predictors, including the product report product code and the reporter's job code, among others. This points to the importance of product-specific characteristics and the reporting source's role in identifying product issues. The negative coefficient for last_four_years_legal_announcementing_firm_most_freq in subset 3 inversely relates the frequency of the most common legal announcements from firms with the outcome, suggesting that a higher frequency of certain types of legal announcements might be linked to a reduced incidence of certain product issues, potentially due to corrective measures taken in response. The accuracy scores from the Lasso regression models for

subsets 1, 2, and 3 are 0.8071235, 0.9829167, and 0.9730068, respectively. The high accuracy scores for subsets 2 and 3, despite the convergence issue in subset 2, suggest that when the model successfully identifies critical predictors, it can significantly predict product outcome incidents. These results underscore the variability in predictor significance across different data segments and highlight the Lasso regression model's utility in pinpointing factors that most strongly influence product safety outcomes.

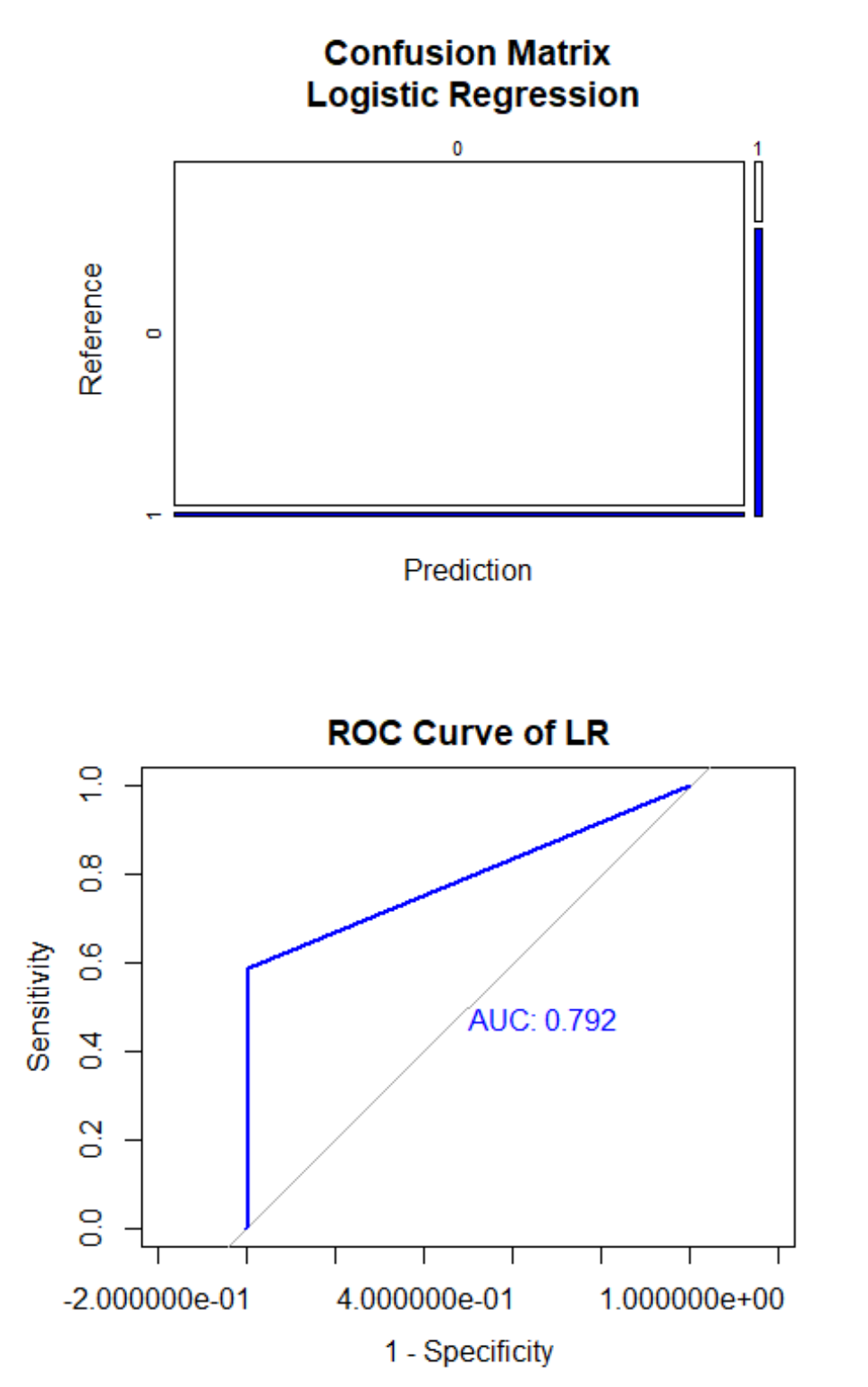
#3. LOGISTIC REGRESSION

```
#logistic regression for subset1
LR_subset1 <- logistic_Regression(formula = Issue_Consequence_new ~
last_two_years_root_cause_description_most_freq+last_two_years_product_quanti
ty_average_max+last_four_years_classification0_num_uniq+last_four_years_legal
_announcementing_firm_num_uniq+last_four_years_product_quantity_average_max+l
ast_four_years_decision_date_max_changes_in_product+manufacturer_contact_addr
ess_1+product.brand_name, train = subset1_sets$train , test =
subset1_sets$test, y_test = subset1_sets$y_test)

## The Accuracy of the Logistic Regression is : 79.97917The accuracy of
Logistic Regression model is 0.7997917

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```



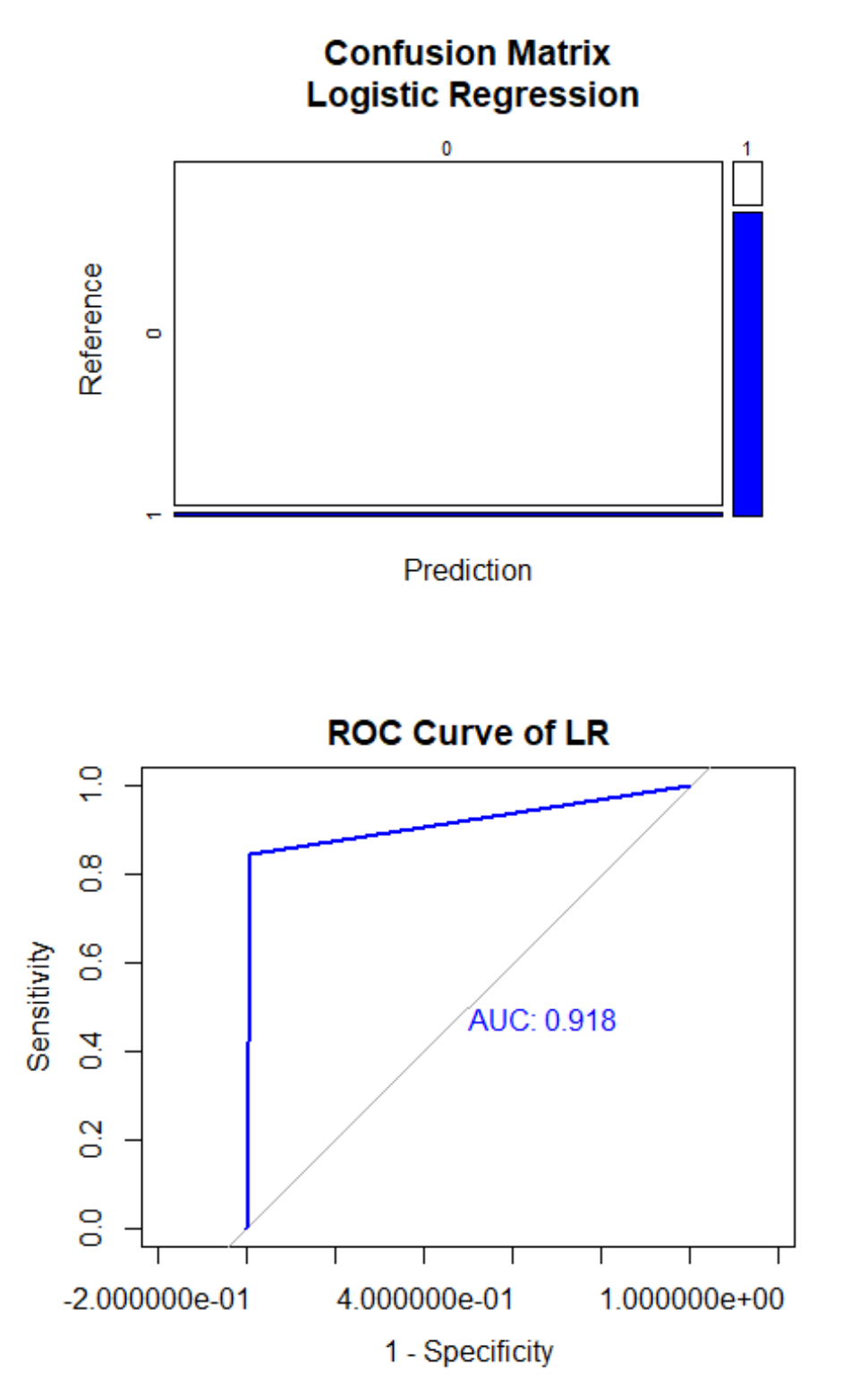

```
LR_subset1

##
## Call:
```

```
## roc.default(response = as.numeric(y_test), predictor =
as.numeric(class_predict))
##
## Data: as.numeric(class_predict) in 3664 controls (as.numeric(y_test) 0) <
1136 cases (as.numeric(y_test) 1).
## Area under the curve: 0.7586

#logistic regression for subset2
LR_subset2 <- logistic_Regression(formula = Issue_Consequence_new ~
last_two_years_root_cause_description_most_freq+last_two_years_product_quanti
ty_average_max+last_four_years_classification0_num_uniq+last_four_years_legal
_announcementing_firm_num_uniq+last_four_years_product_quantity_average_max+l
ast_four_years_decision_date_max_changes_in_product+manufacturer_contact_addr
ess_1+product.brand_name, train = subset2_sets$train , test =
subset2_sets$test, y_test = subset2_sets$y_test)

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

## The Accuracy of the Logistic Regression is : 99.08333The accuracy of
Logistic Regression model is 0.9908333

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```



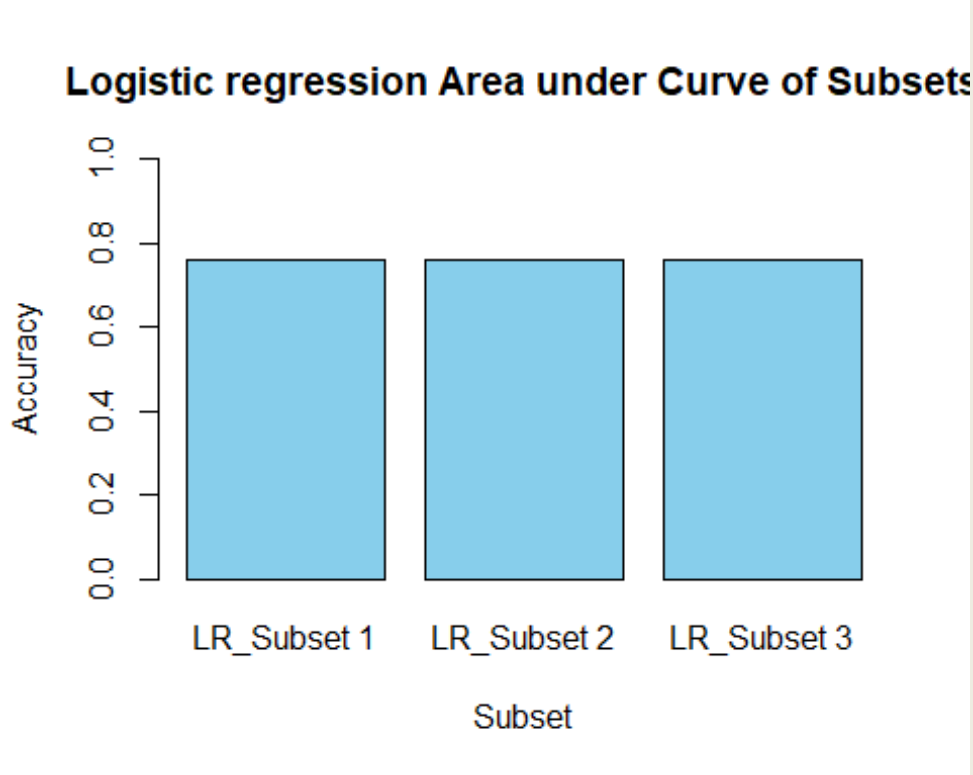

```
LR_subset2

##
## Call:
```

```
## roc.default(response = as.numeric(y_test), predictor =
as.numeric(class_predict))
##
## Data: as.numeric(class_predict) in 4718 controls (as.numeric(y_test) 0) <
82 cases (as.numeric(y_test) 1).
## Area under the curve: 0.7916

#Logistic regression for subset3


LR_subset3 <- logistic_Regression(formula = Issue_Consequence_new
~last_two_years_product_quantity_average_average+
last_four_years_legal_announcementing_firm_most_freq+   product.issue.type+
reporter_job_code   +product.manufacturer_city+ product.field_description
,train = subset3_sets$train, test = subset3_sets$test, y_test =
subset3_sets$y_test)

## The Accuracy of the Logistic Regression is : 98.5699The accuracy of
Logistic Regression model is 0.985699

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```




```
LR_subset3

##
## Call:
```

```
## roc.default(response = as.numeric(y_test), predictor =
as.numeric(class_predict))
##
## Data: as.numeric(class_predict) in 5308 controls (as.numeric(y_test) 0) <
286 cases (as.numeric(y_test) 1).
## Area under the curve: 0.918

accuracy_values <- c(LR_subset1$auc, LR_subset1$auc, LR_subset1$auc)
names(accuracy_values) <- c("LR_Subset 1", "LR_Subset 2", "LR_Subset 3")

# Creating the bar plot
barplot(accuracy_values,
        main="Logistic regression Area under Curve of Subsets",
        xlab="Subset",
        ylab="Accuracy",
        col="skyblue",
        ylim=c(0, 1))  # Assuming accuracy is a proportion between 0 and 1
```



#SHORT SUMMARY OF RESULTS AND INTERPRETATIONS OF THE LOGISTIC REGRESSION

The logistic regression models run on the three subsets of the dataset offer a compelling overview of the model's performance in predicting product issue consequences across different segments.

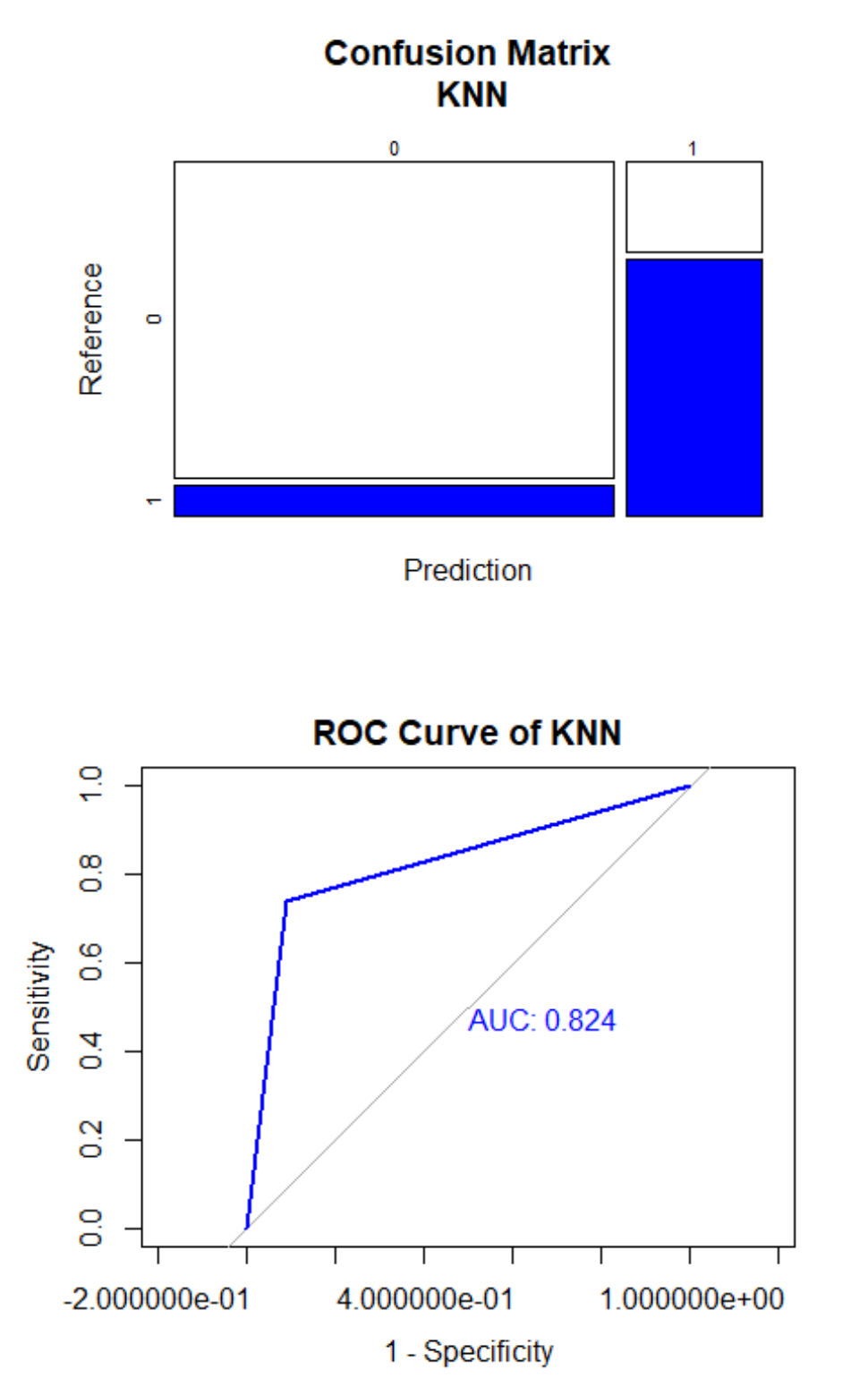
The accuracy for subset 1 is reported at approximately 80%, with an Area Under the Curve (AUC) of 0.763. This indicates a good level of model performance, suggesting that the logistic regression model can distinguish between the controls and cases effectively, albeit with some room for improvement. The relatively lower AUC compared to other subsets might reflect the complexity or noise within this subset, possibly due to a more diverse range of predictors or less clear-cut relationships between predictors and outcomes. Subset 2 Results:

For subset 2, the model achieved a notably high accuracy of 99.08%, accompanied by an AUC of 0.7916. Despite the high accuracy, the AUC, while decent, suggests that while the model is excellent at correctly classifying outcomes, its ability to rank predictions from least to most likely true positives is somewhat less robust. This high accuracy could result from a well-defined separation between cases and controls in this subset, possibly due to a subset of features that are very predictive of the outcome. Subset 3 Results:

The accuracy for subset 3 is similarly high at 98.57%, but with a significantly higher AUC of 0.918, indicating superior model performance not only in classification accuracy but also in the model's ability to rank predictions confidently. This high AUC value suggests that subset 3 contains very predictive features that help distinguish between cases and controls with high reliability.

#KNN

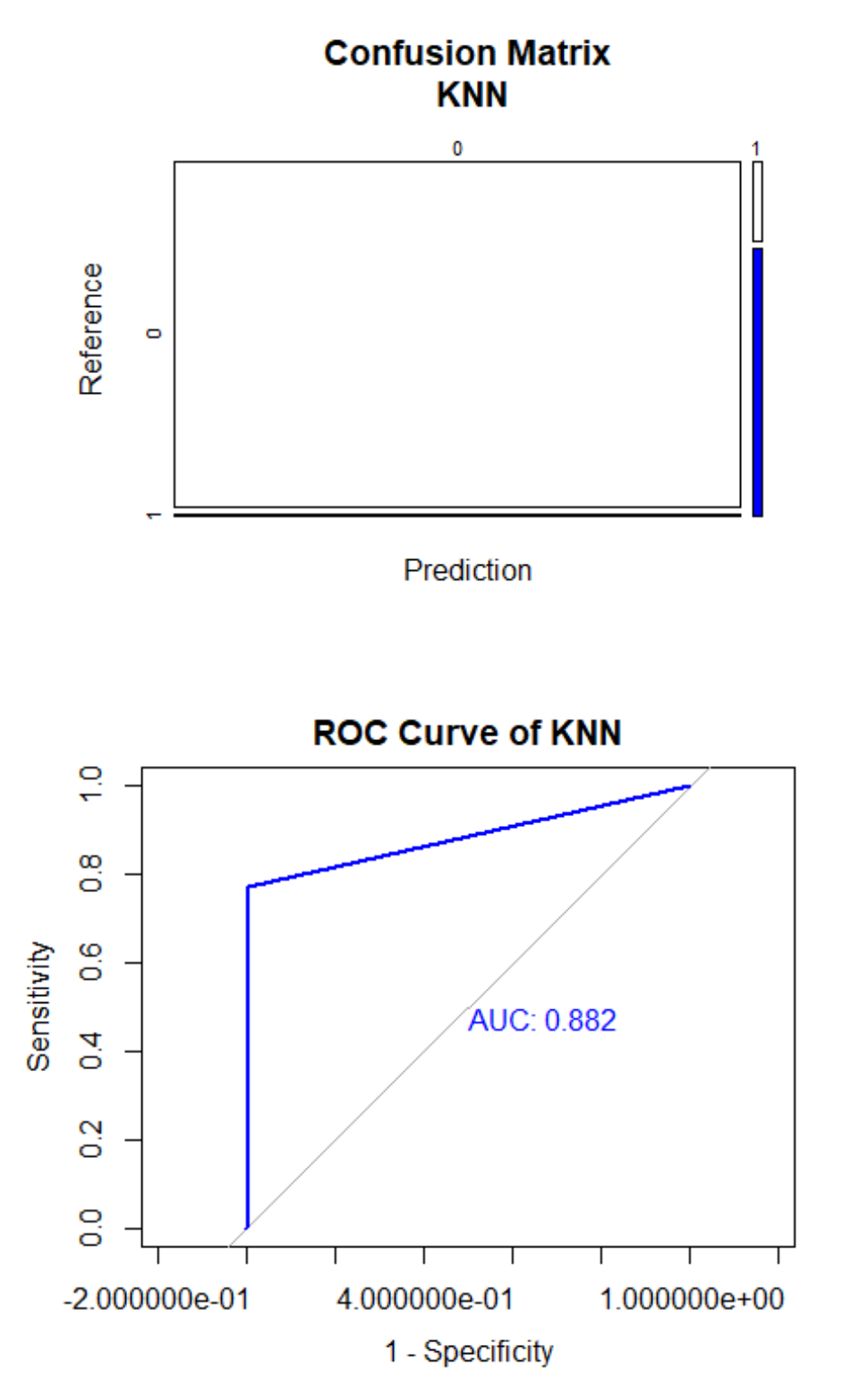
```
#KNN subset-1
knn_subset1 <- knn_classification(train_data = subset1_sets$train, test_data
= subset1_sets$test,
                    formula = Issue_Consequence_new ~
last_two_years_root_cause_description_most_freq+last_two_years_product_quanti
ty_average_max+last_four_years_classification0_num_uniq+last_four_years_legal
_announcementing_firm_num_uniq+last_four_years_product_quantity_average_max+l
ast_four_years_decision_date_max_changes_in_product+manufacturer_contact_addr
ess_1+product.brand_name,k_values = 1:10)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```




```
knn_subset1

## $model
## k-Nearest Neighbors
```

```
##
## 11200 samples
##     8 predictor
##     2 classes: '0', '1'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 10080, 10081, 10080, 10080, 10079, 10079, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.8721437  0.6509224
##    2  0.8697328  0.6444566
##    3  0.8682146  0.6401668
##    4  0.8672329  0.6373359
##    5  0.8661608  0.6342796
##    6  0.8651797  0.6315147
##    7  0.8661621  0.6338202
##    8  0.8647331  0.6303958
##    9  0.8645540  0.6297278
##   10  0.8636620  0.6268778
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
##
## $post_resample_accuracy
## Accuracy
## 0.869375
##
## $confusion_matrix
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3336  328
##          1  299  837
##
##                Accuracy : 0.8694
##                  95% CI : (0.8595, 0.8788)
##     No Information Rate : 0.7573
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6416
##
##  Mcnemar's Test P-Value : 0.2635
##
##             Sensitivity : 0.9177
##             Specificity : 0.7185
##          Pos Pred Value : 0.9105
##          Neg Pred Value : 0.7368
```

```
##              Prevalence : 0.7573
##          Detection Rate : 0.6950
##    Detection Prevalence : 0.7633
##       Balanced Accuracy : 0.8181
##
##        'Positive' Class : 0
##
##
## $roc_curve
##
## Call:
## roc.default(response = test_data$Issue_Consequence_new, predictor =
as.numeric(predictions_knn))
##
## Data: as.numeric(predictions_knn) in 3664 controls
(test_data$Issue_Consequence_new 0) < 1136 cases
(test_data$Issue_Consequence_new 1).
## Area under the curve: 0.8236

 #KNN-Subset 2
knn_subset2 <-knn_classification(train_data = subset2_sets$train, test_data =
subset2_sets$test,
                   formula = Issue_Consequence_new ~
last_two_years_root_cause_description_most_freq+last_two_years_product_quanti
ty_average_max+last_four_years_classification0_num_uniq+last_four_years_legal
_announcementing_firm_num_uniq+last_four_years_product_quantity_average_max+l
ast_four_years_decision_date_max_changes_in_product+manufacturer_contact_addr
ess_1+product.brand_name,k_values = 1:10)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```




```
knn_subset2

## $model
## k-Nearest Neighbors
```

```
##
## 11200 samples
##     8 predictor
##     2 classes: '0', '1'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 10081, 10080, 10080, 10080, 10080, 10080, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9913394  0.7383258
##    2  0.9909821  0.7171708
##    3  0.9902678  0.6926154
##    4  0.9899106  0.6836161
##    5  0.9898212  0.6773807
##    6  0.9891069  0.6617042
##    7  0.9893749  0.6656573
##    8  0.9889285  0.6533037
##    9  0.9891071  0.6567620
##   10  0.9893752  0.6630673
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
##
## $post_resample_accuracy
##  Accuracy
## 0.9922917
##
## $confusion_matrix
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4700   18
##          1   19   63
##
##                Accuracy : 0.9923
##                  95% CI : (0.9894, 0.9946)
##     No Information Rate : 0.9831
##     P-Value [Acc > NIR] : 2.913e-08
##
##                   Kappa : 0.7691
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9960
##             Specificity : 0.7778
##          Pos Pred Value : 0.9962
##          Neg Pred Value : 0.7683
```

```
##              Prevalence : 0.9831
##          Detection Rate : 0.9792
##    Detection Prevalence : 0.9829
##       Balanced Accuracy : 0.8869
##
##        'Positive' Class : 0
##
##
## $roc_curve
##
## Call:
## roc.default(response = test_data$Issue_Consequence_new, predictor =
as.numeric(predictions_knn))
##
## Data: as.numeric(predictions_knn) in 4718 controls
(test_data$Issue_Consequence_new 0) < 82 cases
(test_data$Issue_Consequence_new 1).
## Area under the curve: 0.8822
```

#```{r knn subset-3} #KNN Subset-3 #knn_subset3<-knn_classification(train_data = subset3_sets$train, test_data = subset3_sets$test, formula = Issue_Consequence_new ~ last_two_years_product_quantity_average_average+ #last_four_years_legal_announcementing_firm_most_freq+ product.issue.type+ reporter_job_code +product.manufacturer_city+ product.field_description #+product.product_report_product_code,k_values = 1:10)

#knn_subset3

K-Fold Cross validation

- The model's capacity to generalise to new data that it hasn't seen before is evaluated using cross-validation. K-fold cross-validation enables the model to be trained on numerous different iterations of the data set, lowering the risk of overfitting. Additionally, because it makes use of all of the accessible data for training and testing, it offers a more precise estimate of the model's performance.

- The outcome of cross validation gives performance metrics like accuracy, Kappa, accuracy SD and Kappa SD.

```
kfold_lr_subset1 <- kfold_lr(subset1_sets)

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

kfold_lr_subset1

##   parameter  Accuracy     Kappa AccuracySD KappaSD
## 1      none 0.8804778 0.5266404  0.1510956  0.2212

kfold_lr_subset2 <- kfold_lr(subset2_sets)

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

## 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 in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
```

```
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases
```

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

kfold_lr_subset2

##   parameter  Accuracy     Kappa AccuracySD  KappaSD
## 1      none 0.9090372 0.5676009 0.07823667 0.174205

kfold_lr_subset3 <- kfold_lr(subset3_sets)

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
```

```
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
```

```
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful
cases

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

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

kfold_lr_subset3

##   parameter  Accuracy     Kappa AccuracySD   KappaSD
## 1      none 0.8464083 0.4903728  0.1993498 0.2396865
```

Accuracy and Performance of logistic regression and knn model

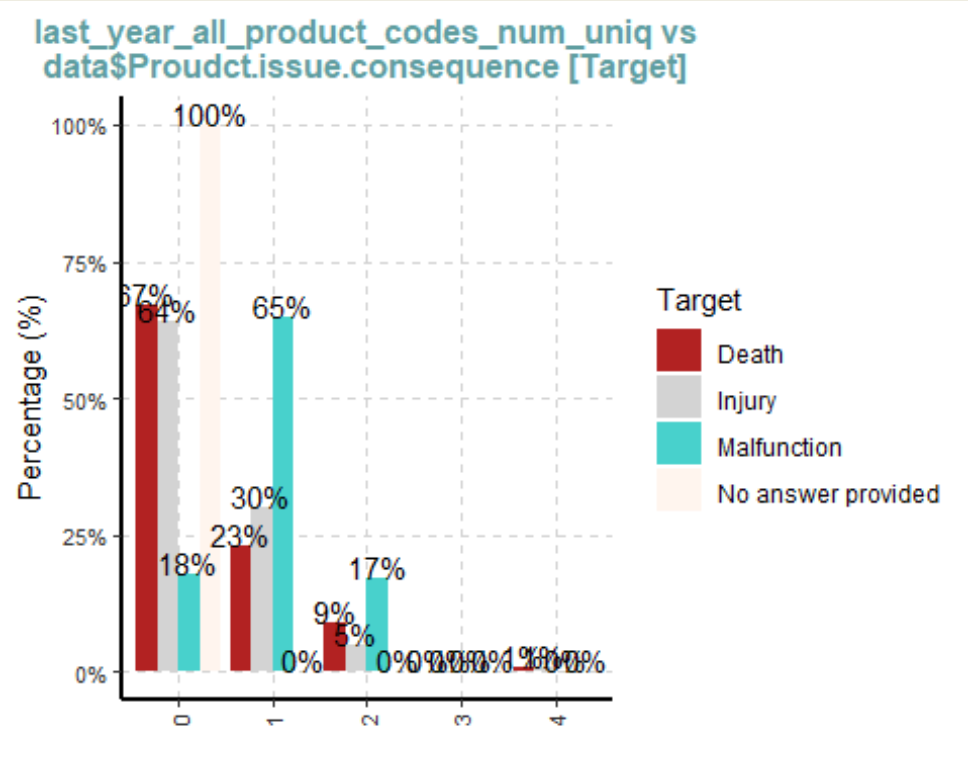
-Logistic Regression shows good performance with accuracies of 79.98%, 99.08%, and 98.57% across the three subsets respectively. The AUC values of 0.7586, 0.7916, and 0.918 indicate a decent ability to distinguish between the classes, with the third subset showing exceptional discrimination capability. -KNN (k=1) demonstrates superior accuracy, especially in subsets 2 and 3, with accuracy rates closely matching or surpassing those of logistic regression. The AUC for subset 1 is 0.8236, which surpasses the logistic regression model for the same subset, indicating a better model performance in terms of both accuracy and the ability to discriminate between the positive and negative classes.

#Preferred Model Given the results, the preference between logistic regression and KNN might depend on the specific requirements of the task at hand:

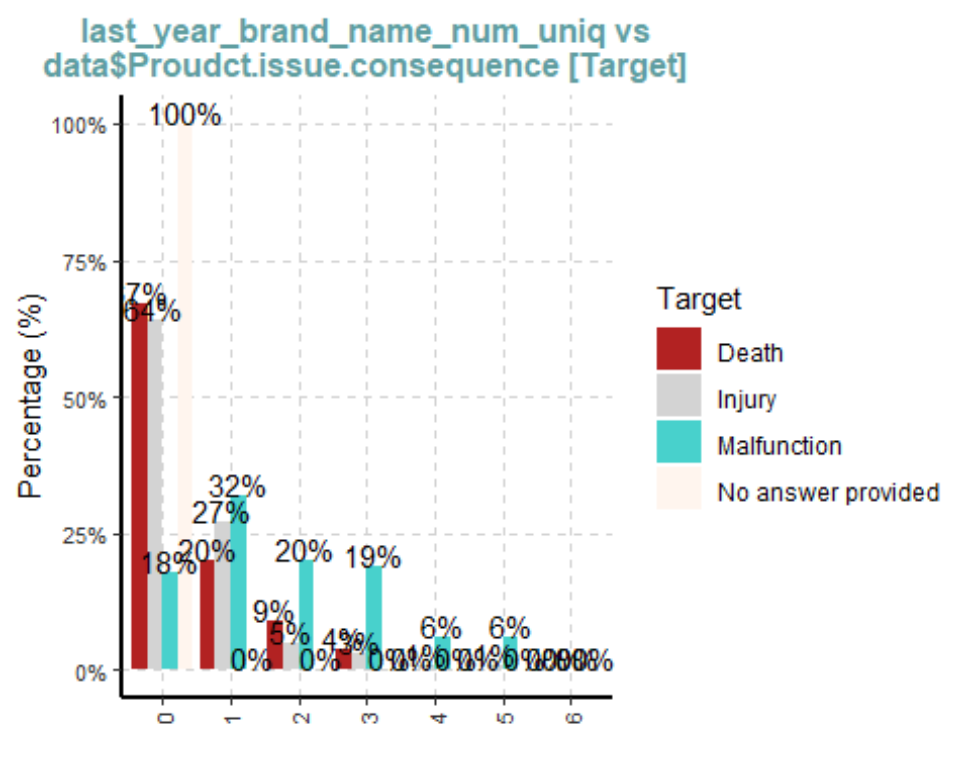
-If interpretability and understanding the influence of predictors are paramount, logistic regression would be preferable despite its slightly lower performance in some subsets. It offers a balance between accuracy and the ability to provide insights into the data. If the primary goal is maximizing accuracy and the ability to discriminate between classes, especially in more complex datasets, KNN could be favored, particularly with an appropriate choice of k. However, it requires careful preprocessing and consideration of the distance metric. In conclusion, logistic regression offers a good balance of interpretability and performance, making it suitable for scenarios where understanding the impact of predictors is crucial. On the other hand, KNN may be preferred in applications where the highest possible accuracy is the sole objective, assuming the complexity and lower interpretability of the model are acceptable trade-offs.

APPENDIX

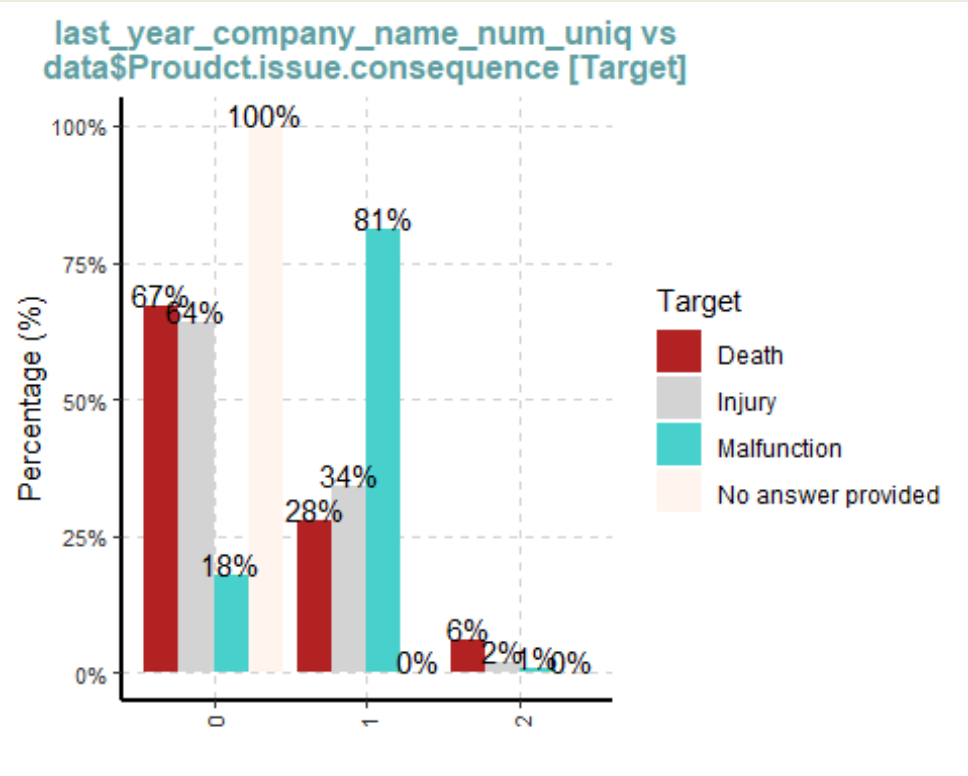
```
SmartEDA::ExpCatViz(data = last_year_1, target =
"data$Proudct.issue.consequence" )

## [[1]]
```



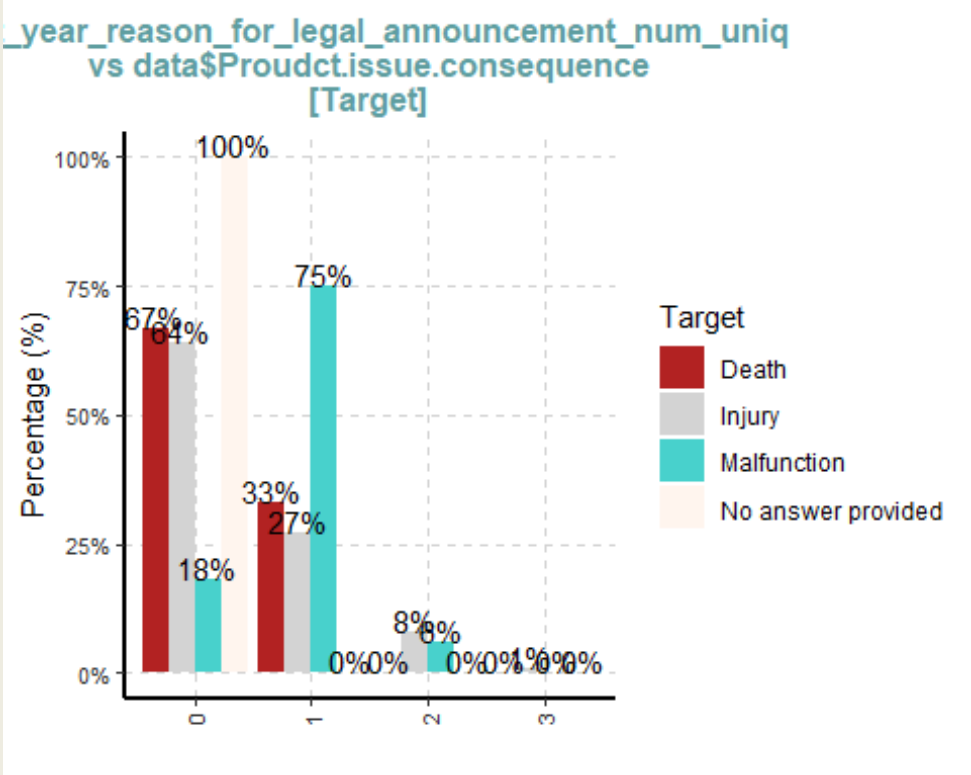

```
##
## [[2]]
```


last_year_brand_name_num_uniq vs
data$Proudct.issue.consequence [Target]
Percentage (%)
100%
75%
50%
25%
0%
0
1
2
3
4
5
6
Target
Death
Injury
Malfunction
No answer provided

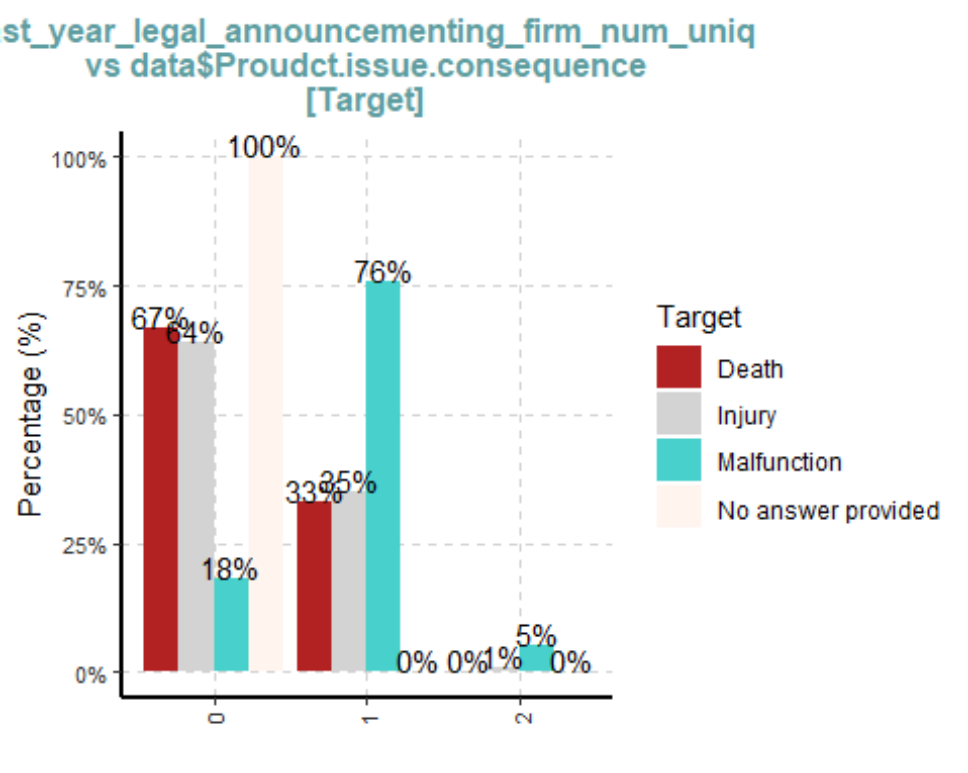
```
##
## [[3]]
```


last_year_company_name_num_uniq vs
data$Proudct.issue.consequence [Target]
Percentage (%)
100%
75%
50%
25%
0%
0
1
2
Target
Death
Injury
Malfunction
No answer provided

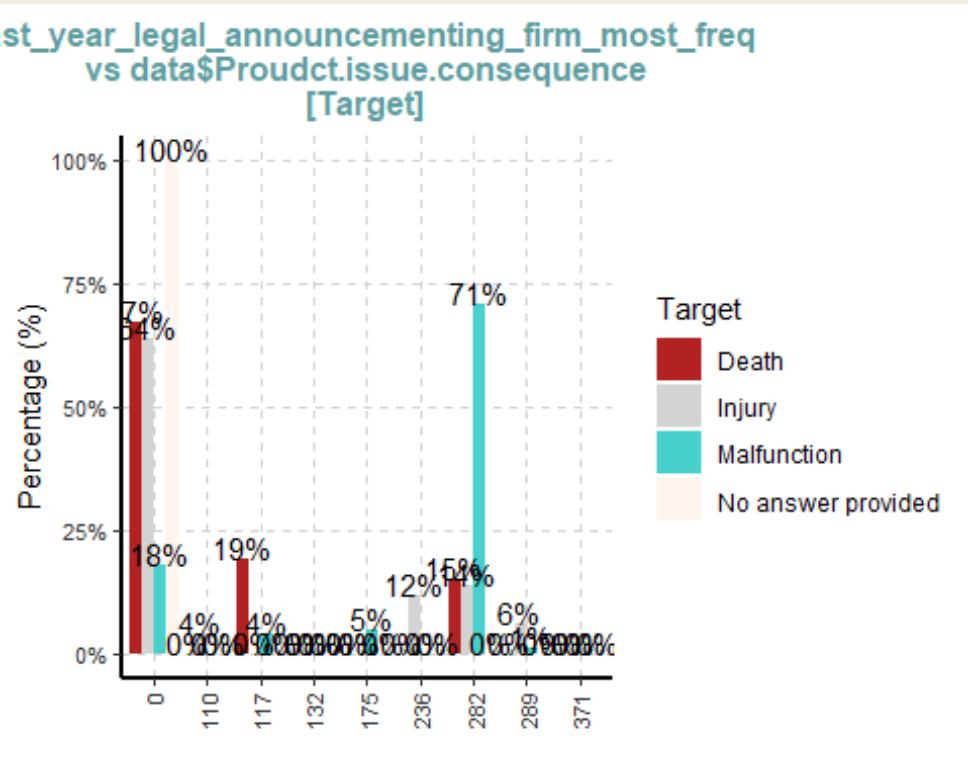
```
##
## [[4]]
```


_year_reason_for_legal_announcement_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
100%
75%
50%
25%
0%
67%
64%
18%
100%
33%
27%
75%
0%
0%
8%
6%
0%
0%
1%
0%
0%
0
1
2
3
Target
Death
Injury
Malfunction
No answer provided

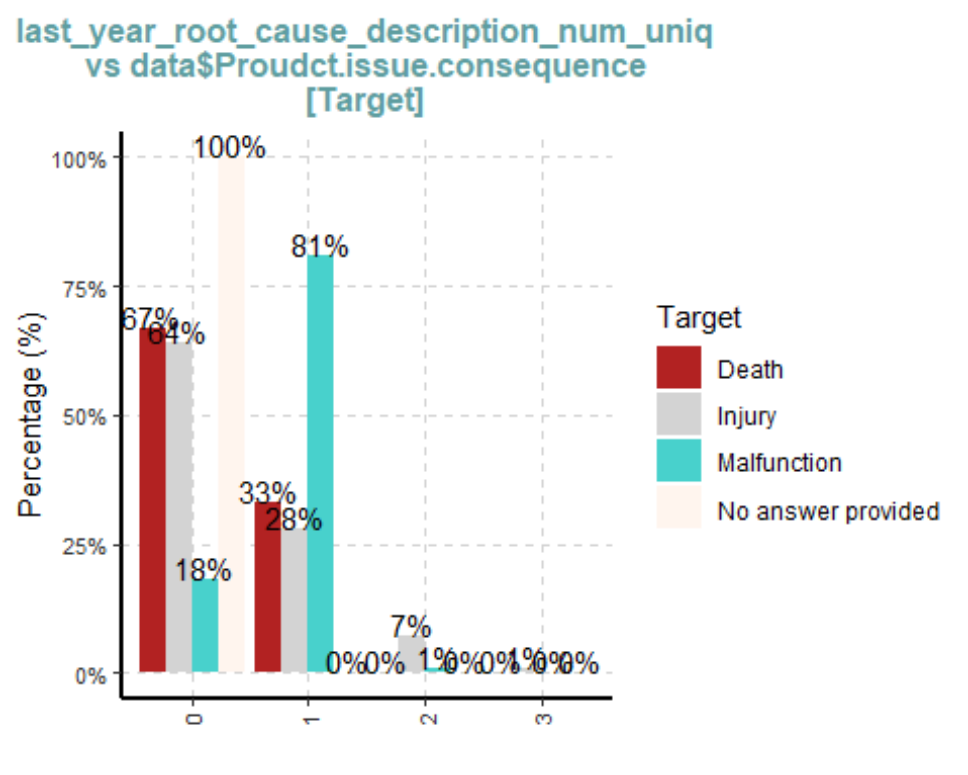
```
##
## [[5]]
```


st_year_legal_announcementing_firm_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
100%
75%
50%
25%
0%
67%
64%
100%
18%
33%
35%
76%
0%
0%
1%
5%
0%
0
1
2
Target
Death
Injury
Malfunction
No answer provided

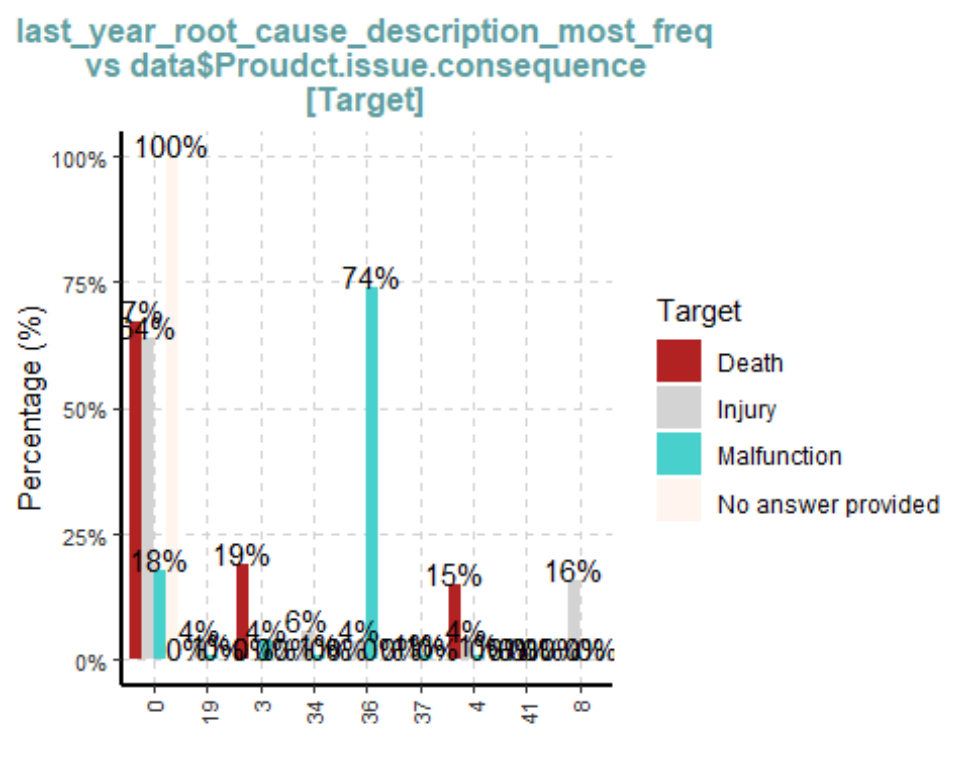
```
##
## [[6]]
```


st_year_legal_announcementing_firm_most_freq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
100%
75%
50%
25%
0%
100%
18%
19%
4%
5%
12%
15%
71%
6%
0
110
117
132
175
236
282
289
371
Target
Death
Injury
Malfunction
No answer provided

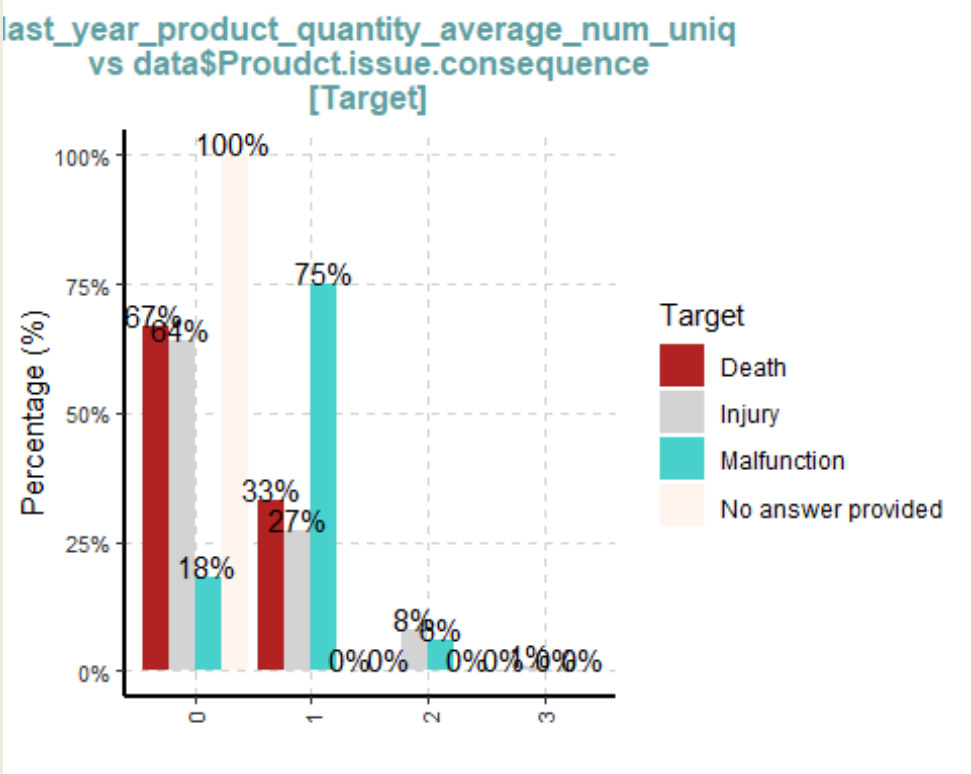
```
##
## [[7]]
```


last_year_root_cause_description_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
100%
75%
50%
25%
0%
67%
64%
18%
100%
33%
28%
81%
7%
1%
0%
0%
0%
0%
0%
0%
0
1
2
3
Target
Death
Injury
Malfunction
No answer provided

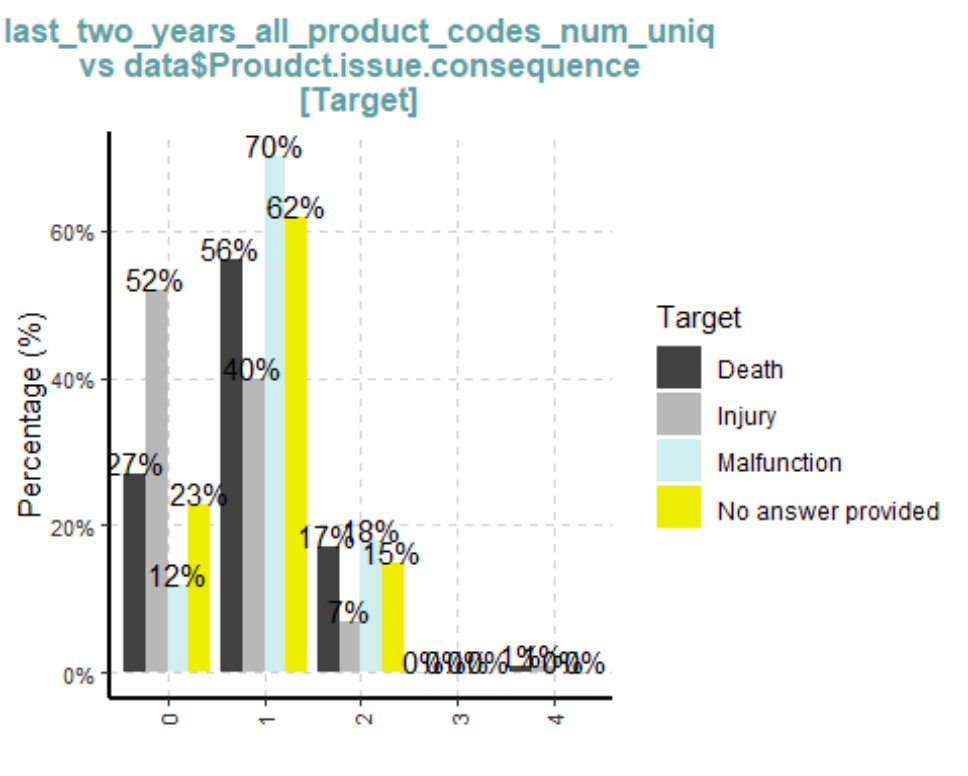
```
##
## [[8]]
```


last_year_root_cause_description_most_freq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
100%
75%
50%
25%
0%
100%
74%
54%
18%
19%
15%
16%
4%
6%
4%
0
19
3
34
36
37
4
41
8
Target
Death
Injury
Malfunction
No answer provided

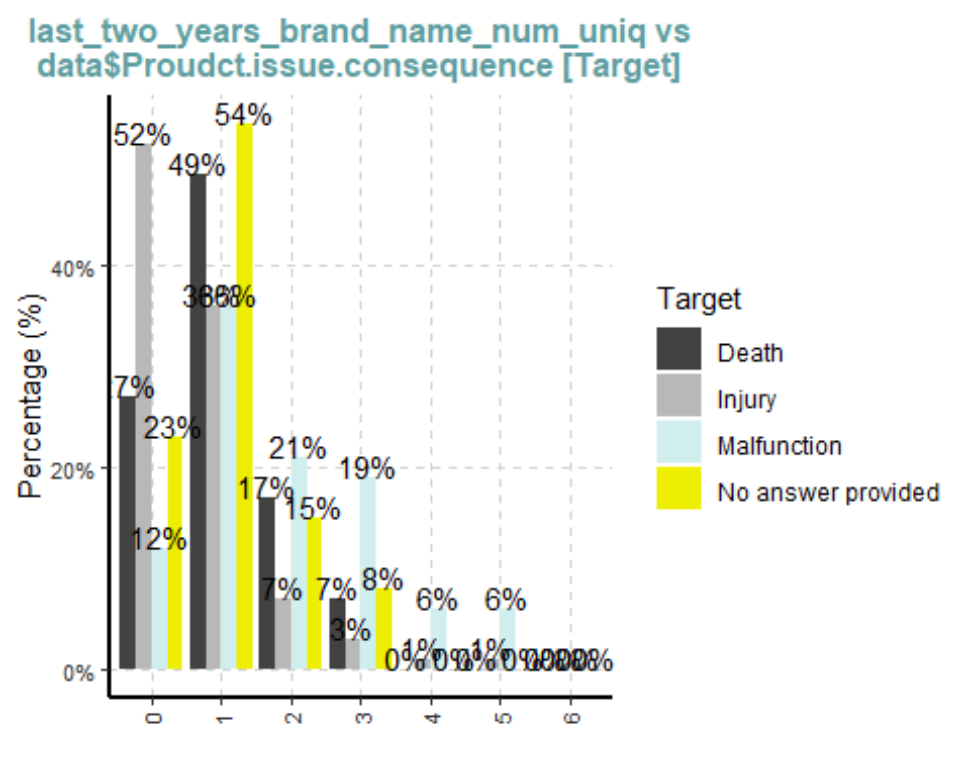
```
##
## [[9]]
```


last_year_product_quantity_average_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
100%
75%
50%
25%
0%
100%
75%
67%
64%
33%
27%
18%
8%
8%
0%
0%
0%
0%
0
1
2
3
Target
Death
Injury
Malfunction
No answer provided

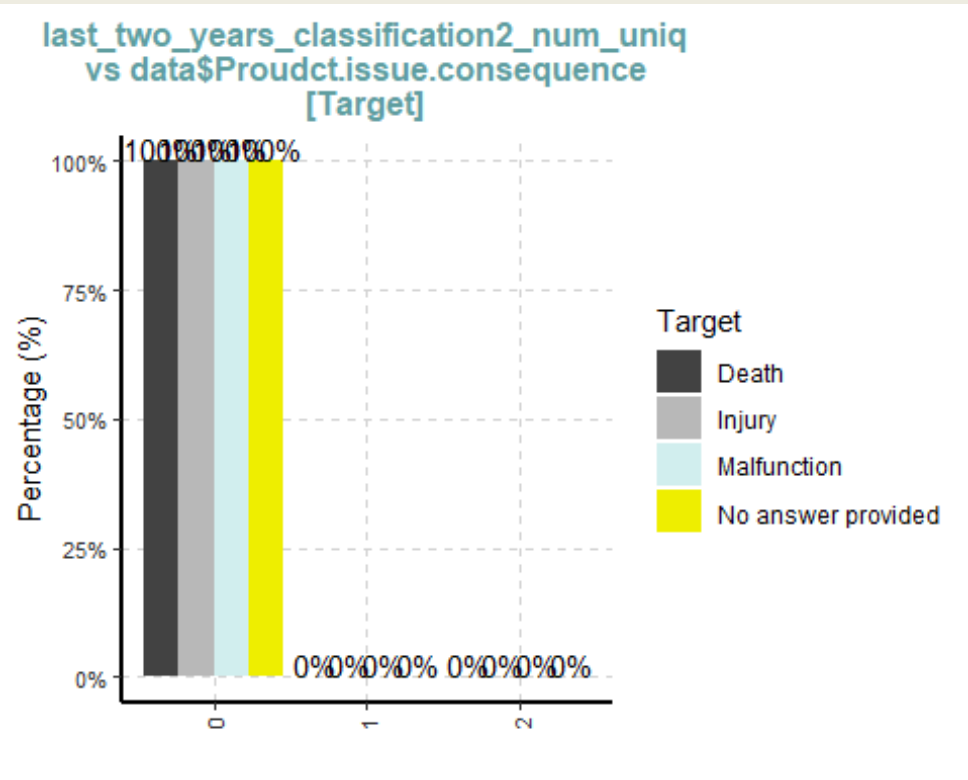
```
SmartEDA::ExpCatViz(data = last_year_2, target =
"data$Proudct.issue.consequence")

## [[1]]
```



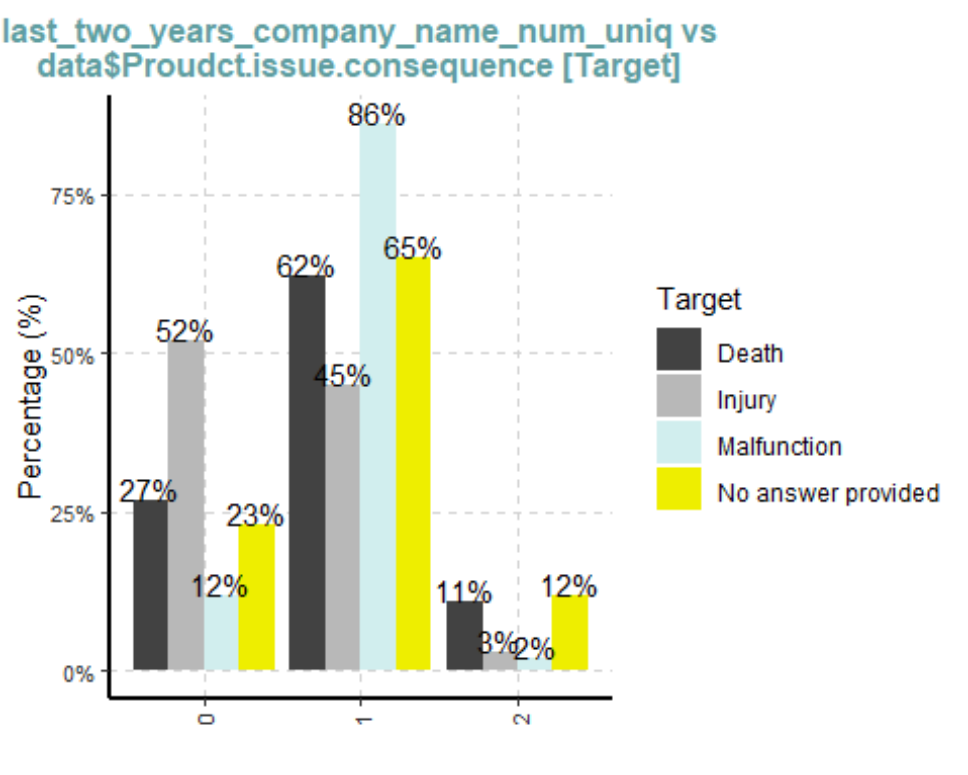

```
##
## [[2]]
```


last_two_years_brand_name_num_uniq vs data$Proudct.issue.consequence [Target]
Percentage (%)
40%
20%
0%
52%
49%
54%
7%
23%
12%
21%
19%
17%
15%
7%
7%
8%
6%
6%
3%
0
1
2
3
4
5
6
Target
Death
Injury
Malfunction
No answer provided

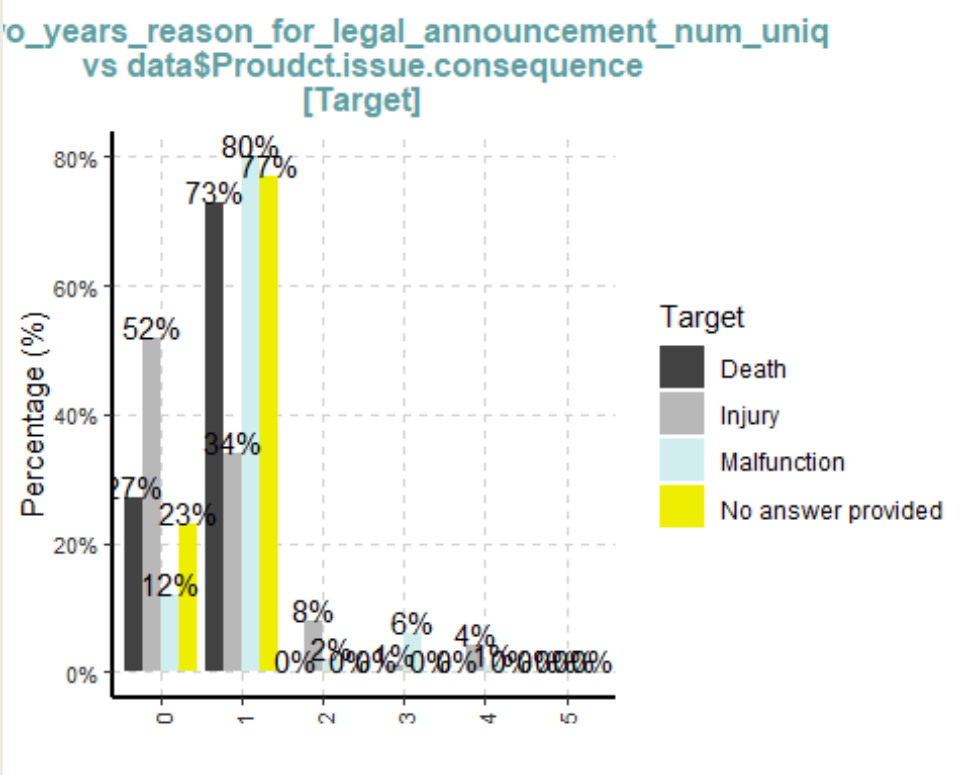
```
##
## [[3]]
```


last_two_years_classification2_num_uniq vs data$Proudct.issue.consequence [Target]
Percentage (%)
100%
75%
50%
25%
0%
0
1
2
Target
Death
Injury
Malfunction
No answer provided

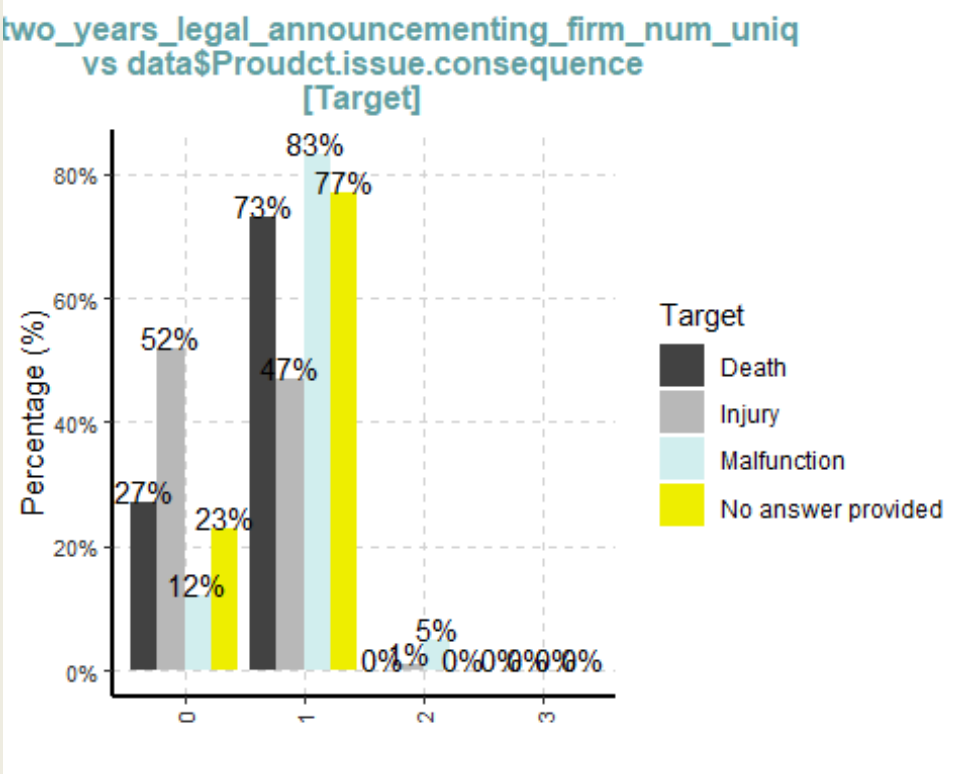
```
##
## [[4]]
```


last_two_years_company_name_num_uniq vs
data$Proudct.issue.consequence [Target]
Percentage (%)
75%
50%
25%
0%
27%
52%
12%
23%
62%
45%
86%
65%
11%
3%
2%
12%
0
1
2
Target
Death
Injury
Malfunction
No answer provided

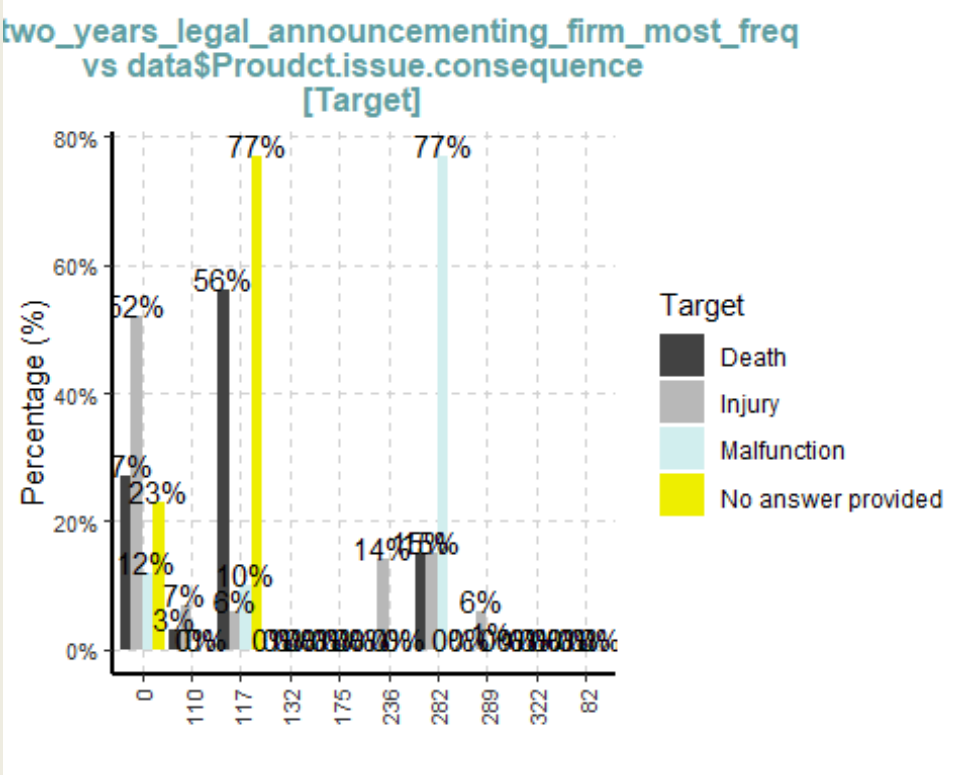
```
##
## [[5]]
```


o_years_reason_for_legal_announcement_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
80%
60%
40%
20%
0%
52%
27%
23%
12%
73%
80%
77%
34%
8%
2%
6%
4%
0
1
2
3
4
5
Target
Death
Injury
Malfunction
No answer provided

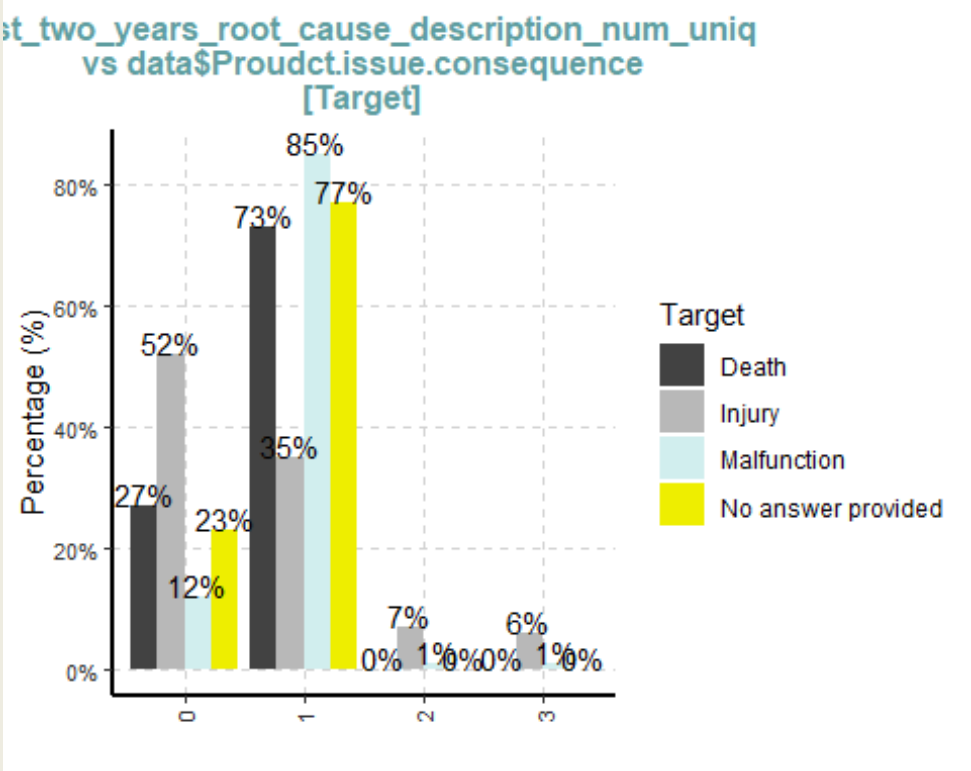
```
##
## [[6]]
```


two_years_legal_announcementing_firm_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
80%
60%
40%
20%
0%
27%
52%
12%
23%
73%
47%
83%
77%
1%
5%
0%
0
1
2
3
Target
Death
Injury
Malfunction
No answer provided

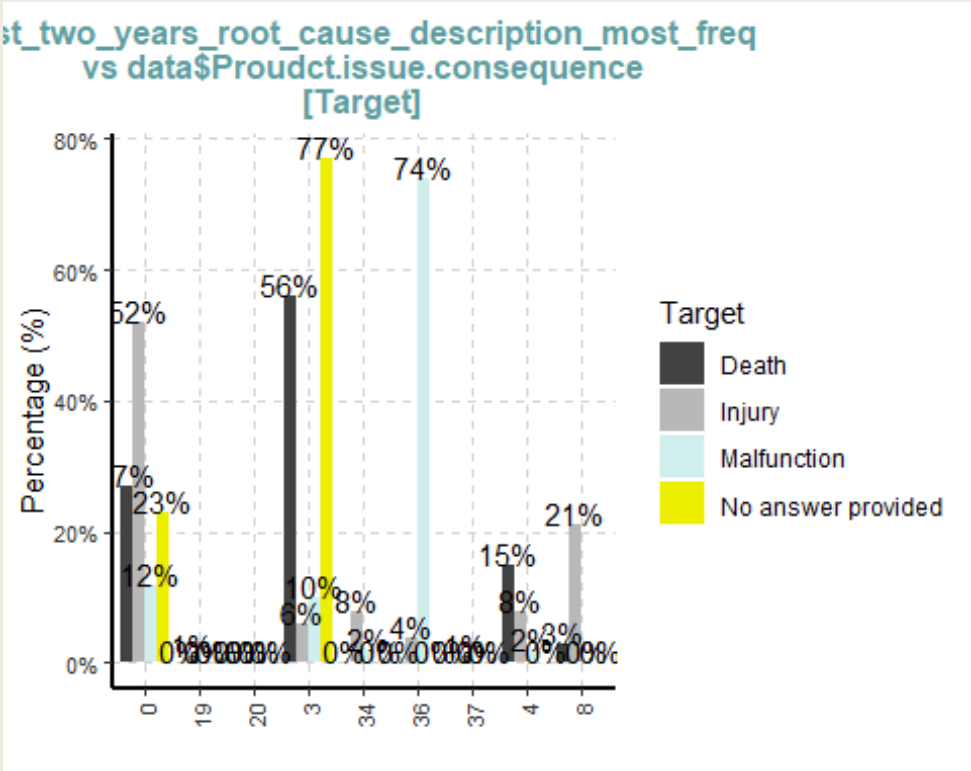
```
##
## [[7]]
```


two_years_legal_announcementing_firm_most_freq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
80%
60%
40%
20%
0%
77%
77%
56%
52%
23%
12%
10%
7%
6%
3%
14%
6%
0
110
117
132
175
236
282
289
322
82
Target
Death
Injury
Malfunction
No answer provided

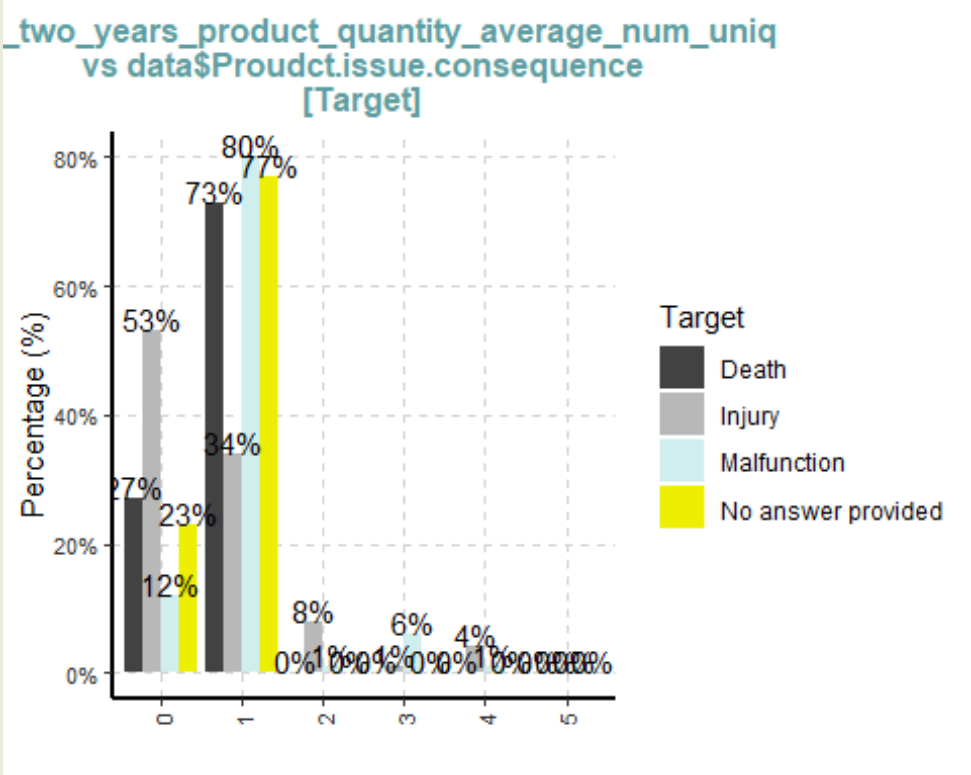
```
##
## [[8]]
```


st_two_years_root_cause_description_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
80%
60%
40%
20%
0%
85%
77%
73%
52%
35%
27%
23%
12%
7%
6%
1%
0%
0%
0%
1%
0%
0
1
2
3
Target
Death
Injury
Malfunction
No answer provided

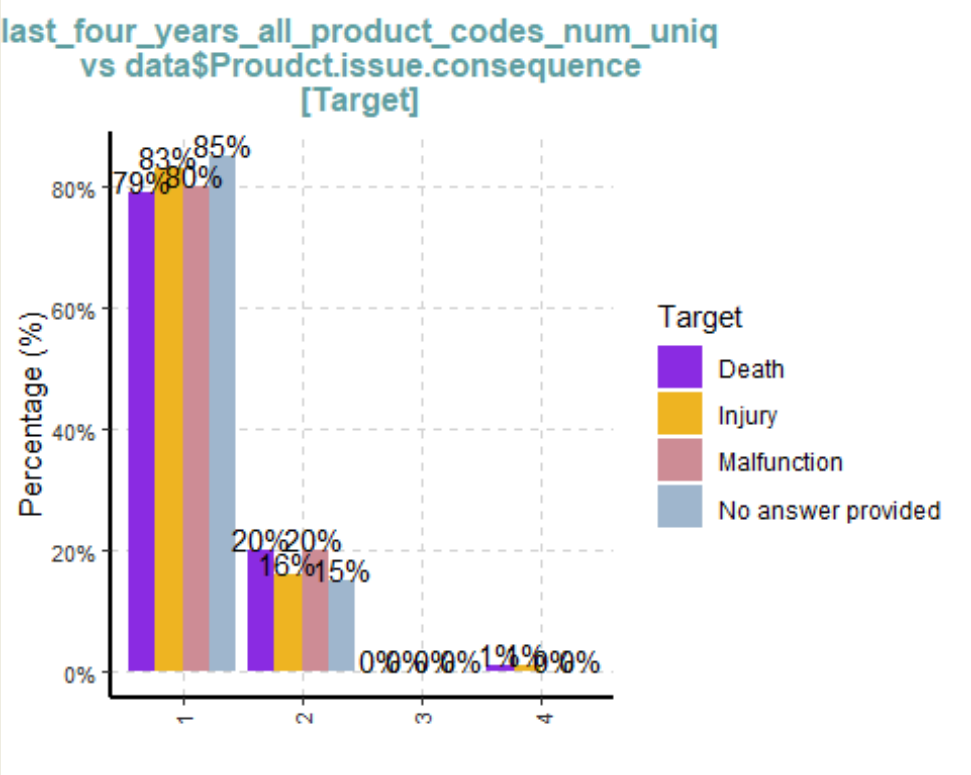
```
##
## [[9]]
```


st_two_years_root_cause_description_most_freq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
80%
60%
40%
20%
0%
77%
74%
56%
52%
7%
23%
12%
21%
15%
10%
8%
6%
8%
4%
2%
2%
0
19
20
3
34
36
37
4
8
Target
Death
Injury
Malfunction
No answer provided

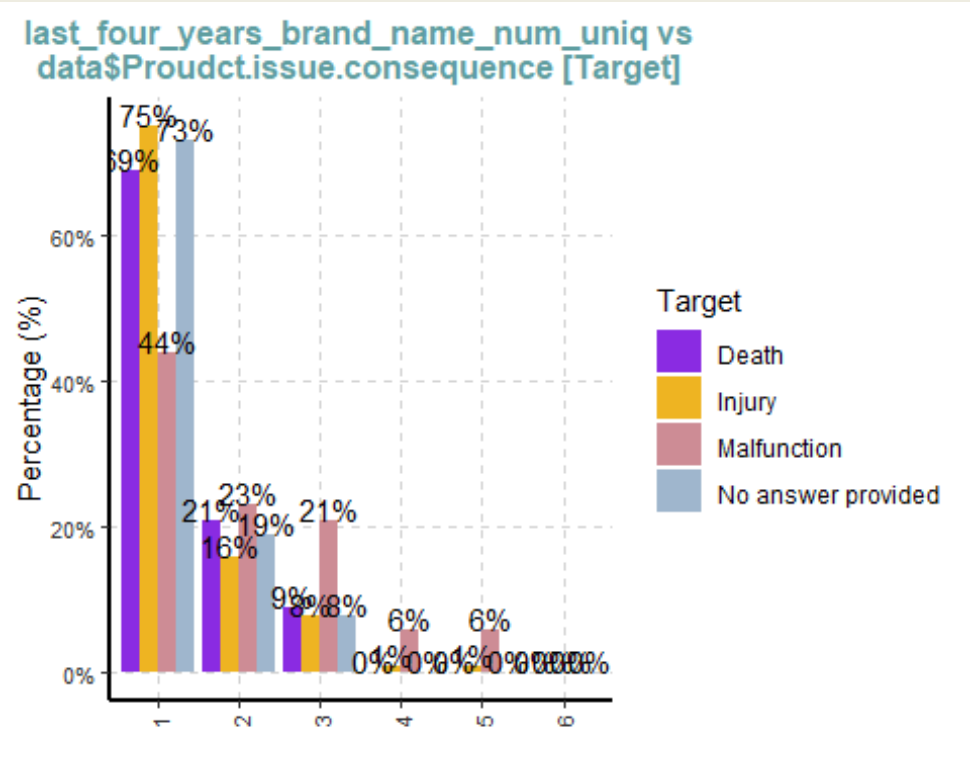
```
##
## [[10]]
```



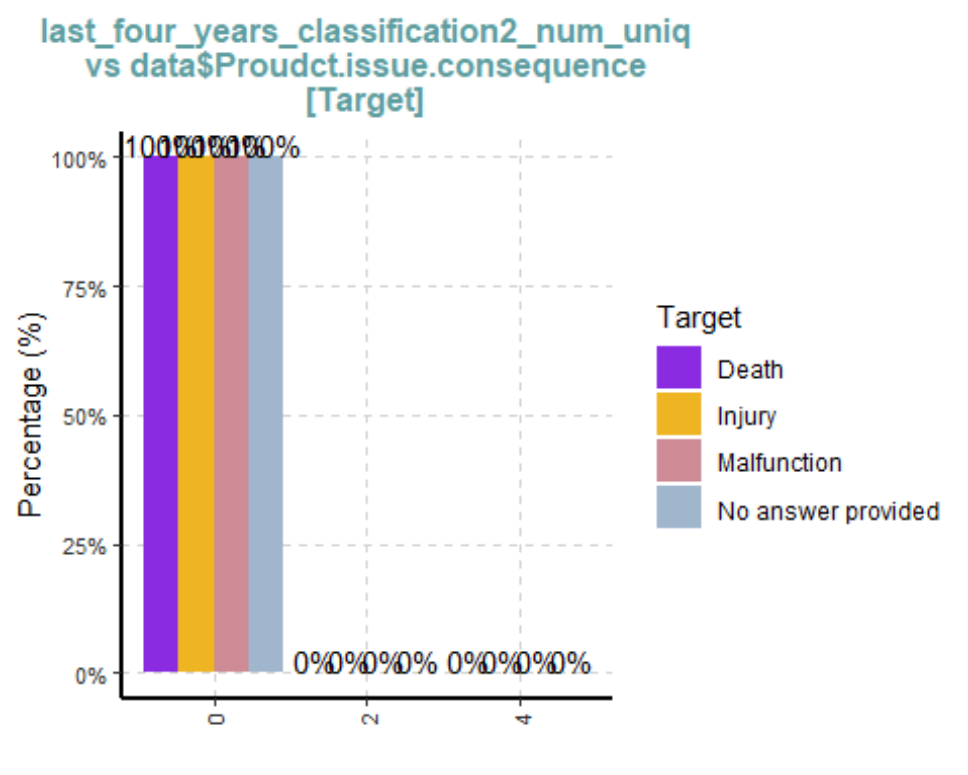

```
SmartEDA::ExpCatViz(data = last_year_4, target =
"data$Proudct.issue.consequence")

## [[1]]
```


last_four_years_all_product_codes_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
80%
60%
40%
20%
0%
79%
83%
80%
85%
20%
16%
20%
15%
1%
1
2
3
4
Target
Death
Injury
Malfunction
No answer provided

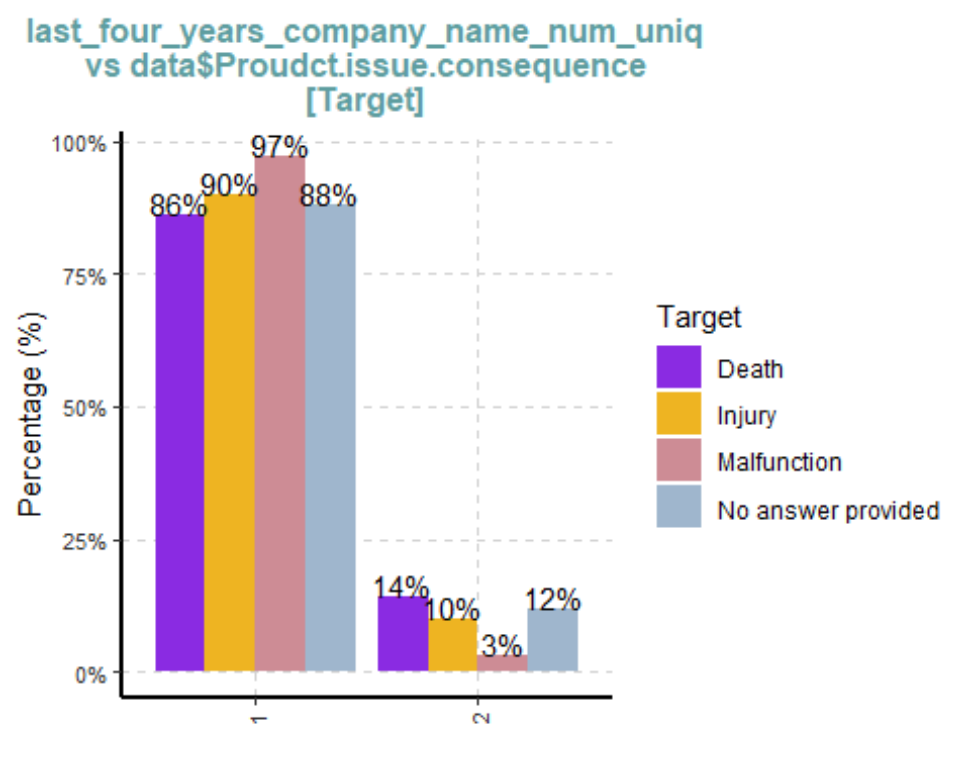
```
##
## [[2]]
```


last_four_years_brand_name_num_uniq vs
data$Proudct.issue.consequence [Target]
Percentage (%)
60%
40%
20%
0%
75%
73%
44%
21%
16%
23%
19%
21%
9%
8%
6%
6%
1
2
3
4
5
6
Target
Death
Injury
Malfunction
No answer provided

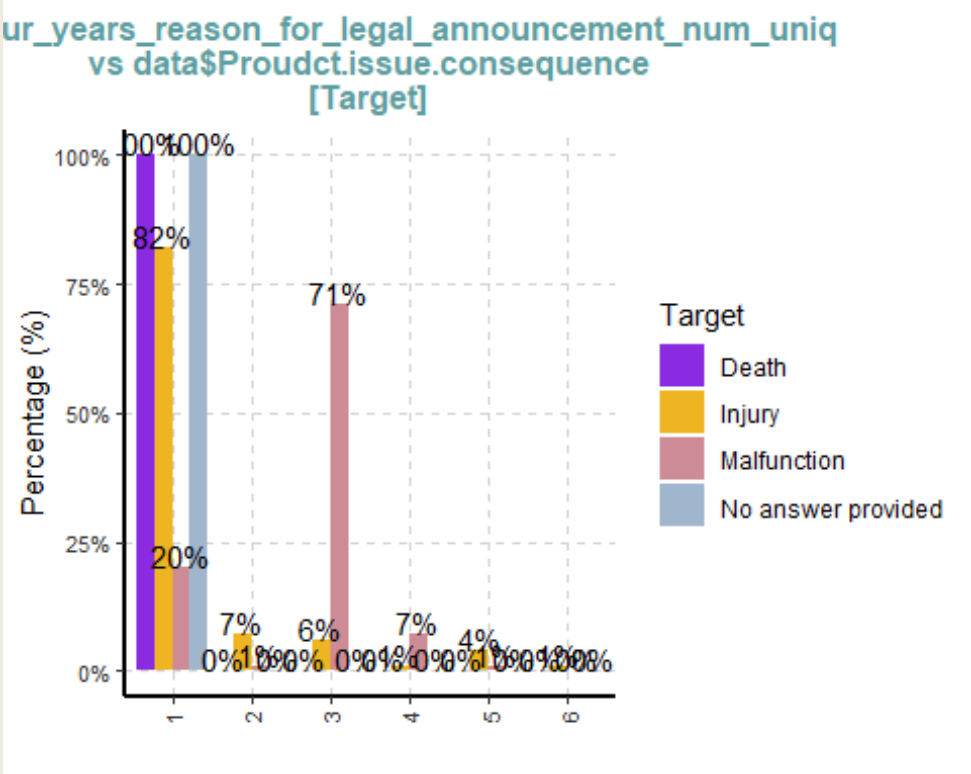
```
##
## [[3]]
```


last_four_years_classification2_num_uniq
vs data$Proudct.issue.consequence
[Target]
100%
75%
50%
25%
0%
Percentage (%)
0%0%0%0% 0%0%0%0%
0
2
4
Target
Death
Injury
Malfunction
No answer provided

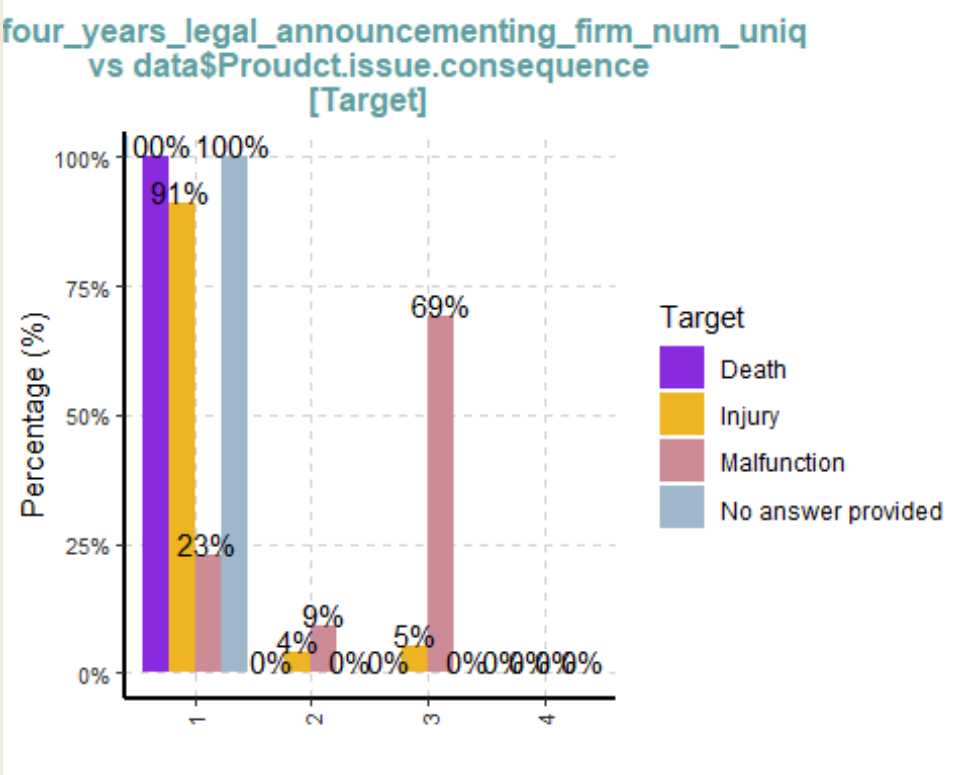
```
##
## [[4]]
```


last_four_years_company_name_num_uniq
vs data$Proudct.issue.consequence
[Target]
100%
75%
50%
25%
0%
Percentage (%)
86%
90%
97%
88%
14%
10%
3%
12%
1
2
Target
Death
Injury
Malfunction
No answer provided

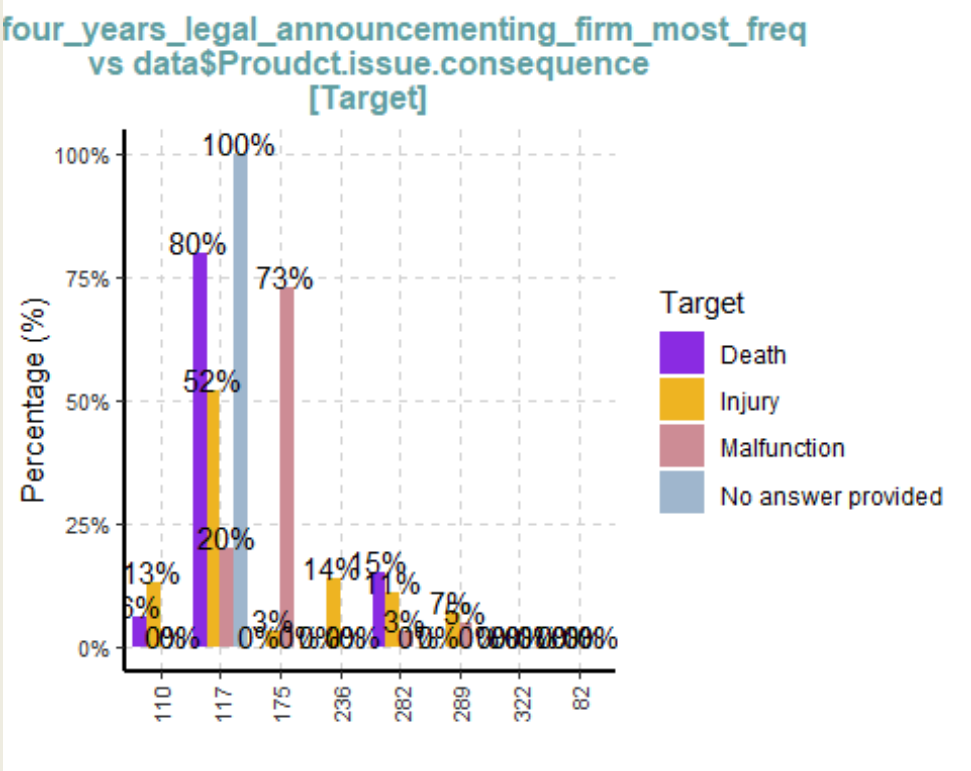
```
##
## [[5]]
```


ur_years_reason_for_legal_announcement_num_uniq
vs data$Proudct.issue.consequence
[Target]
100%
75%
50%
25%
0%
Percentage (%)
82%
20%
71%
7%
6%
7%
4%
1
2
3
4
5
6
Target
Death
Injury
Malfunction
No answer provided

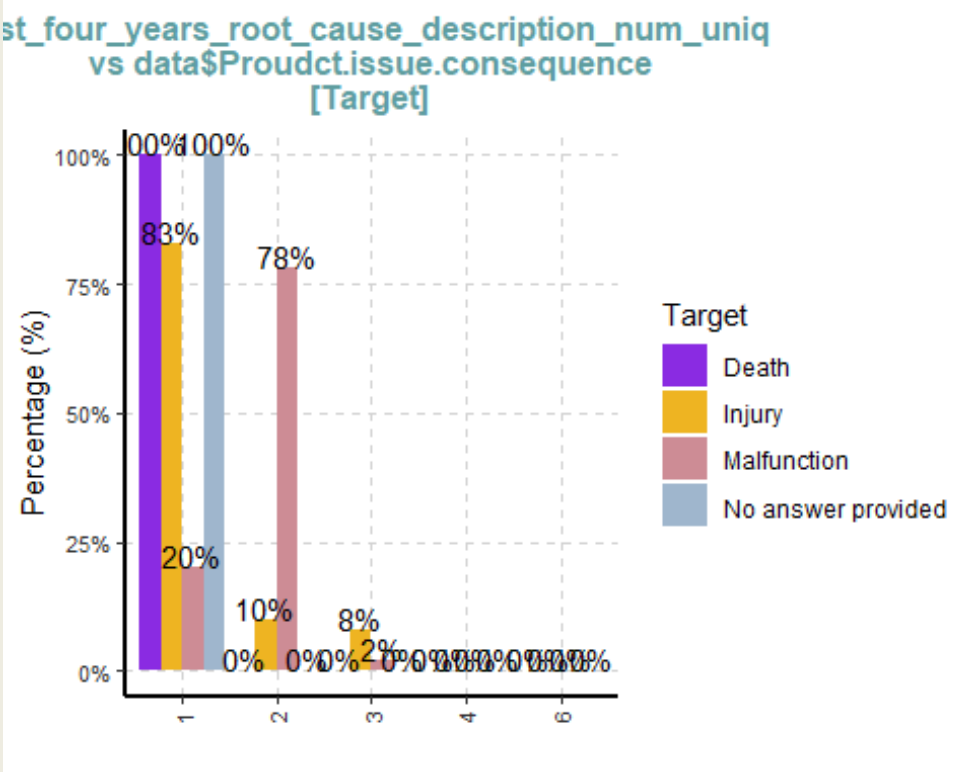
```
##
## [[6]]
```


four_years_legal_announcementing_firm_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
100%
75%
50%
25%
0%
00% 100%
91%
23%
0%
4%
9%
0%
0%
5%
69%
0%
0%
0%
0%
0%
1
2
3
4
Target
Death
Injury
Malfunction
No answer provided

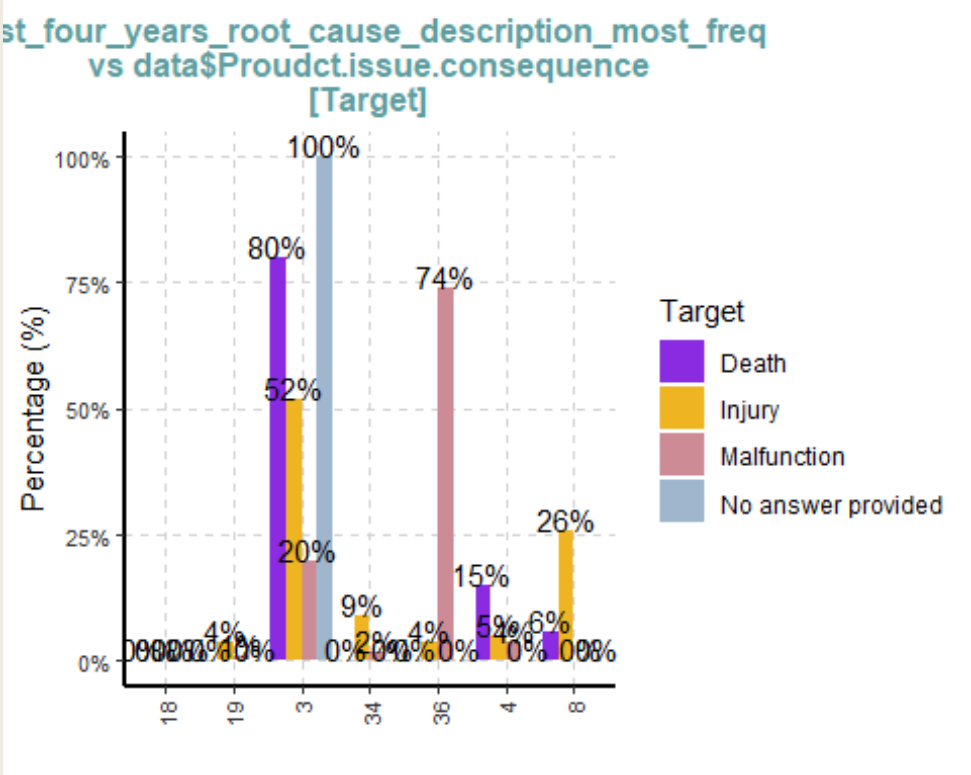
```
##
## [[7]]
```


four_years_legal_announcementing_firm_most_freq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
100%
75%
50%
25%
0%
110
117
175
236
282
289
322
82
Target
Death
Injury
Malfunction
No answer provided

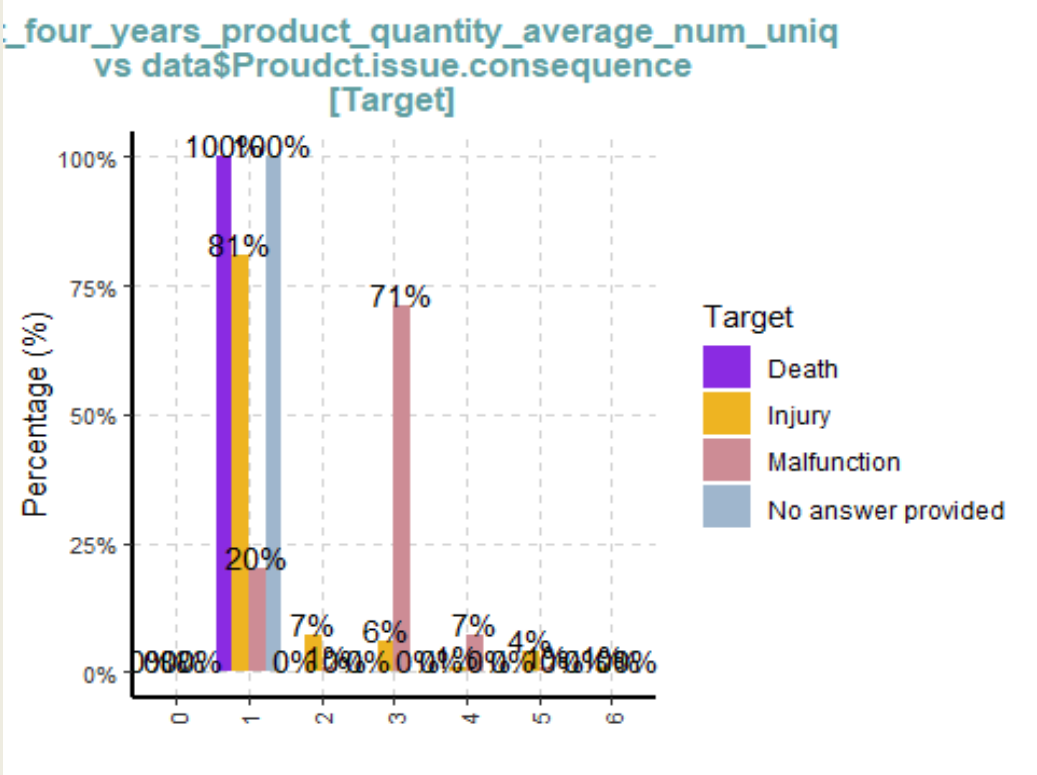
```
##
## [[8]]
```


st_four_years_root_cause_description_num_uniq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
100%
75%
50%
25%
0%
1
2
3
4
6
Target
Death
Injury
Malfunction
No answer provided

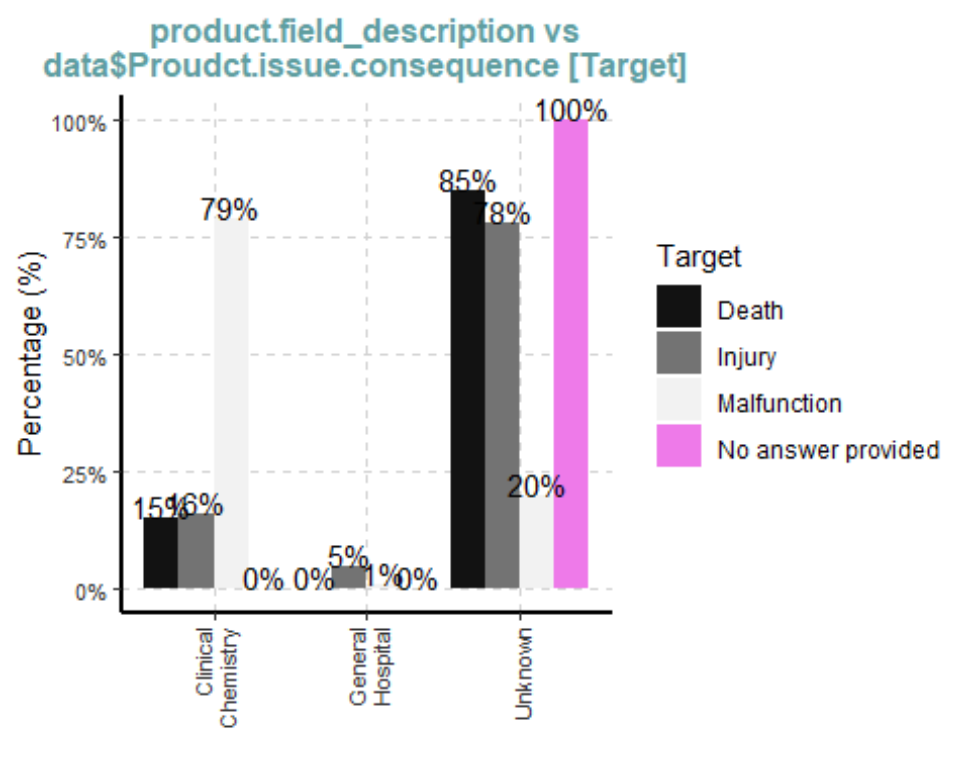
```
##
## [[9]]
```


st_four_years_root_cause_description_most_freq
vs data$Proudct.issue.consequence
[Target]
Percentage (%)
100%
75%
50%
25%
0%
80%
52%
20%
100%
4%
9%
2%
74%
4%
15%
5%
26%
6%
0%
18
19
3
34
36
4
8
Target
Death
Injury
Malfunction
No answer provided

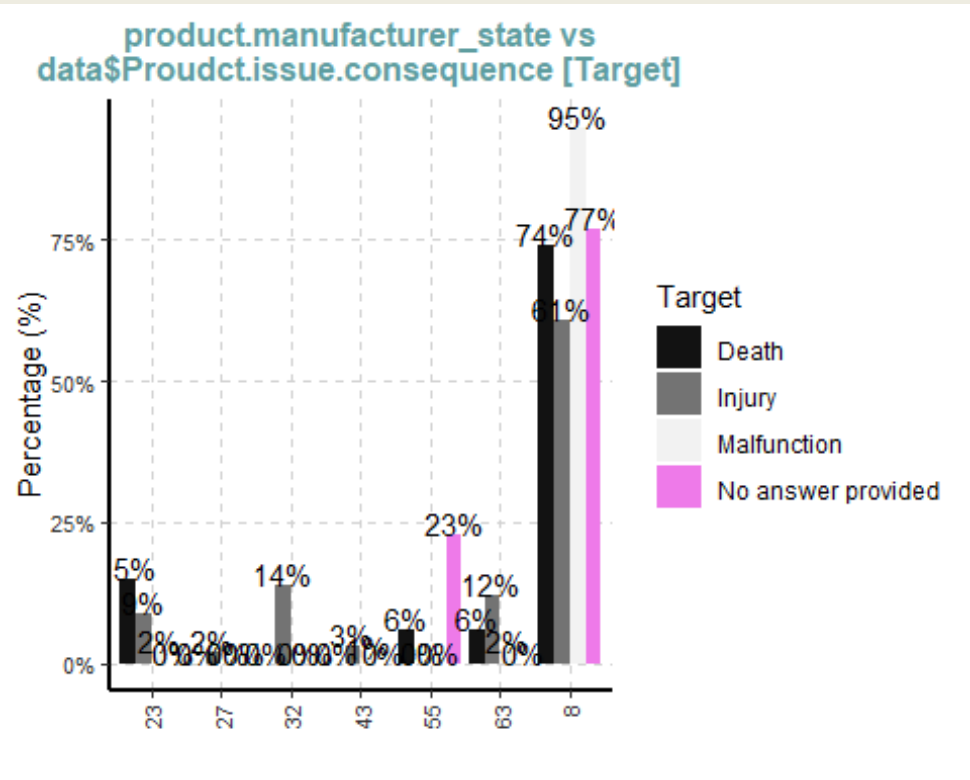
```
##
## [[10]]
```



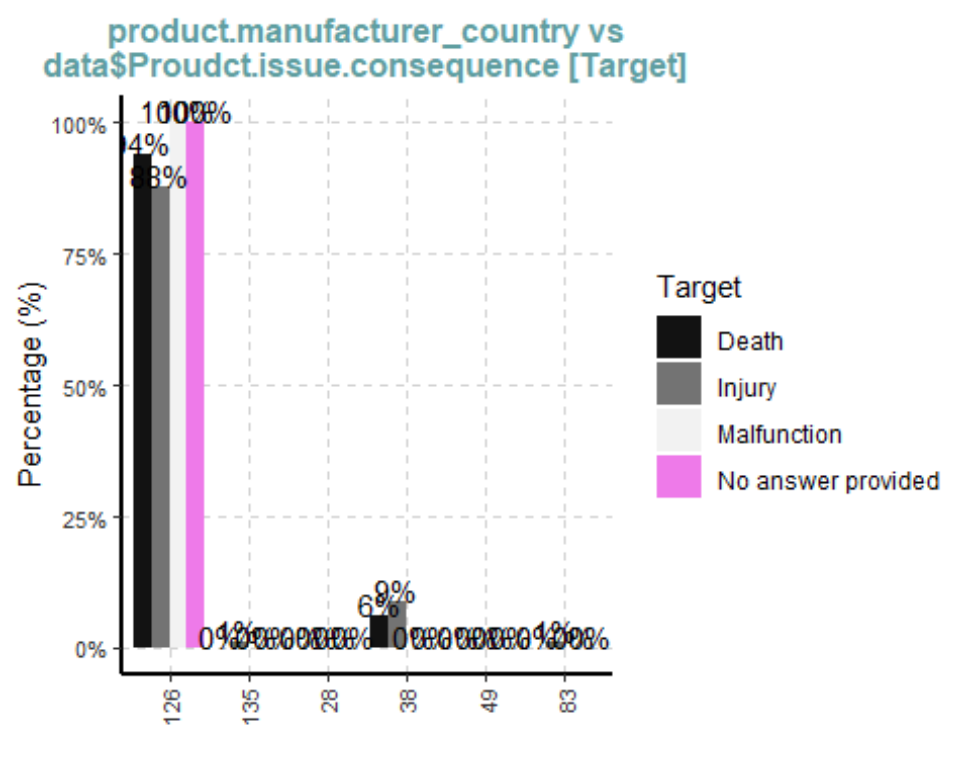

```
SmartEDA::ExpCatViz(data = product_1, target =
"data$Proudct.issue.consequence")

## [[1]]
```


product.field_description vs
data$Proudct.issue.consequence [Target]
Percentage (%)
100%
75%
50%
25%
0%
79%
85%
78%
100%
20%
5%
0% 0%
1%
0%
Clinical Chemistry
General Hospital
Unknown
Target
Death
Injury
Malfunction
No answer provided

```
##
## [[2]]
```


product.manufacturer_state vs
data$Proudct.issue.consequence [Target]
Percentage (%)
75%
50%
25%
0%
95%
74%
77%
23%
14%
12%
6%
6%
5%
2%
23
27
32
43
55
63
8
Target
Death
Injury
Malfunction
No answer provided

```
##
## [[3]]
```


product.manufacturer_country vs
data$Proudct.issue.consequence [Target]
Percentage (%)
100%
75%
50%
25%
0%
9%
6%
126
135
28
38
49
83
Target
Death
Injury
Malfunction
No answer provided