Predicts Fraud

CPE 213 Data Models

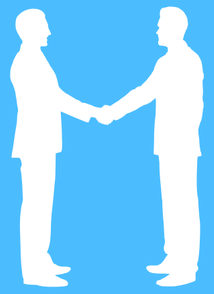
PROBLEMS

MONEY

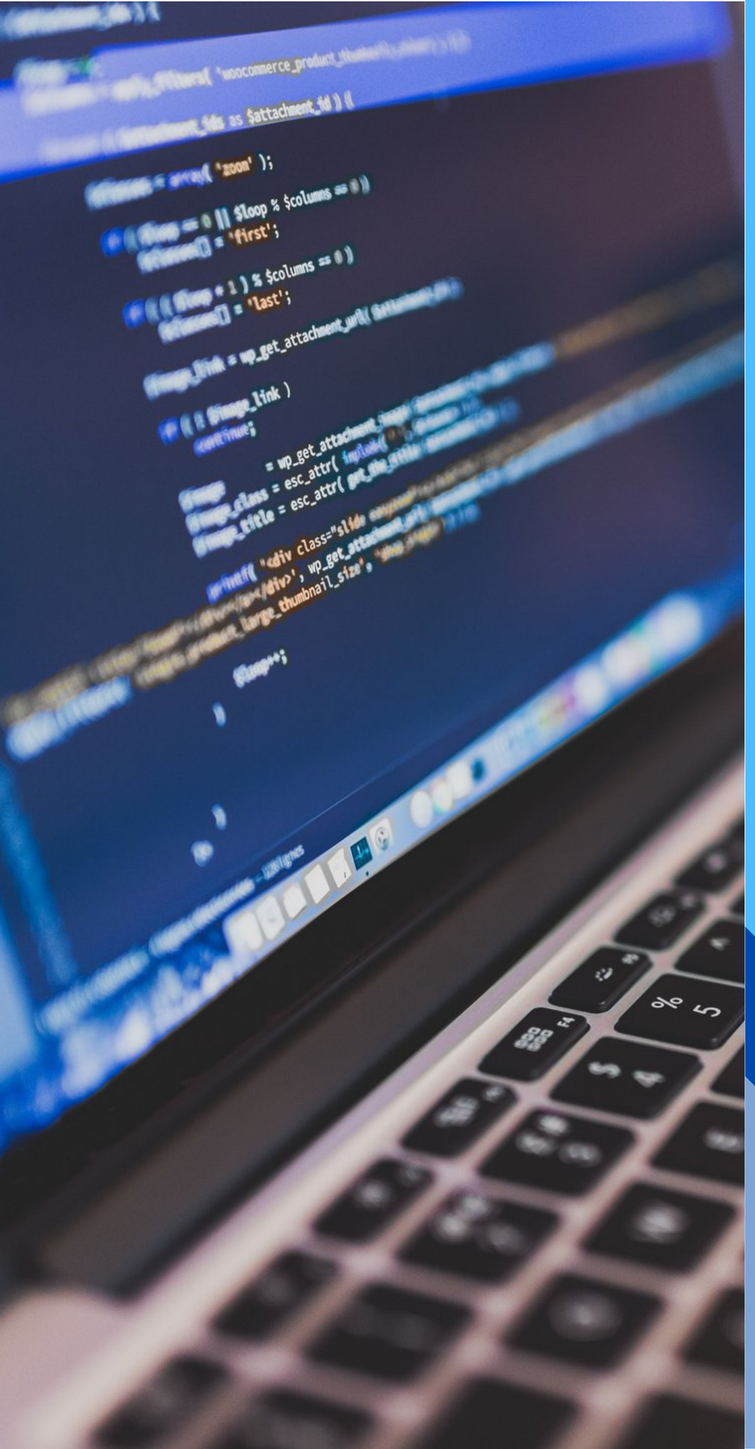
Prevent unnecessary loss

TRUST

Find correct fraud



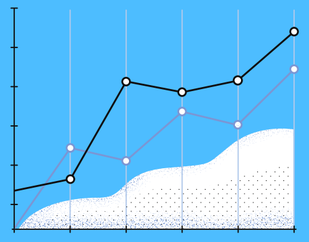
PROBLEMS



OBJECTIVES

KNOWLEDGE

To use knowledge from study course



ANALYTIC

Knowledge along with Business

Data description & Preparation

DATA DESCRIPTION

REGION	Customer region details
STATE	Current location of customer
AREA	Area in Urban and Rural
CITY	Customers current located city
CONSUMER_PROFILE	Customer's work profile Personal/Business
PRODUCT_CATEGORY	Product category • Household • Entertainment
SERVICE_CENTRE	7 Service centers

PRODUCT_AGE	Duration of the product purchased by customer
PURCHASED_FROM	From where product is purchased • Internet • Dealer • Manufacture
CALL_DETAILS	Call duration in minute
PURPOSE	Consumer Purpose • Claim • Complaint • Other
FRAUD	• 1 fradulent claim • 0 Genuin claim
CLAIM_VALUE	Customer's claim amount in Rs. (Rupee)

PRODUCT_TYPE

- Type of the product TV/AC

failure of Compressor in AC	AC_1001_ISSUE	TV_2001_ISSUE	failure of power supply in TV
failure of Condenser Coil in AC	AC_1002_ISSUE	TV_2002_ISSUE	failure of Inverter in TV
failure of Evaporator Coil in AC	AC_1003_ISSUE	TV_2003_ISSUE	failure of Motherboard in TV

ISSUE CATEGORY
0 - NO ISSUE / NO COMPONENT
1 - REPAIR
2 - REPLACEMENT

PREPARATION

```
df <- read.csv('claims.csv', stringsAsFactors = TRUE)

names(df)[1] <- "Case_ID"

df$Fraud <- factor(ifelse(df$Fraud == 0, "No", "Yes"))
df$Purpose <- as.character(df$Purpose)
df$Purpose <- factor(if_else(df$Purpose == "claim" , "Claim" , df$Purpose))
df$Service_Centre <- factor(as.character(df$Service_Centre))
df$AC_1001_Issue <- factor(as.character(df$AC_1001_Issue))
df$AC_1002_Issue <- factor(as.character(df$AC_1002_Issue))
df$AC_1003_Issue <- factor(as.character(df$AC_1003_Issue))
df$TV_2001_Issue <- factor(as.character(df$TV_2001_Issue))
df$TV_2002_Issue <- factor(as.character(df$TV_2002_Issue))
df$TV_2003_Issue <- factor(as.character(df$TV_2003_Issue))

claims_df <- df # No pivot

df <- df %>% pivot_longer(AC_1001_Issue:TV_2003_Issue,
                          names_to = "Issue",
                          values_to = "Issue_type",
                          values_transform = as.character())

df <- df[df$Issue_type != 0,]

str(claims_df)
str(df)
```

PREPARATION

MAIN DATA FRAME

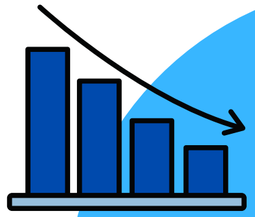
```
> str(claims_df)
'data.frame':   8341 obs. of  21 variables:
 $ Case_ID          : int  7957 1396 7582 5824 4086 6721 1185 3954 8820 58231 ...
 $ Region           : Factor w/ 8 levels "East","North",..: 3 1 8 8 7 3 6 1 7 5 ...
 $ State            : Factor w/ 22 levels "Andhra Pradesh",..: 2 4 1 6 12 22 13 11 1 1 ...
 $ Area             : Factor w/ 2 levels "Rural","Urban": 1 2 1 1 1 2 1 2 2 1 ...
 $ City             : Factor w/ 27 levels "Agartala","Ahmedabad",..: 9 17 10 2 12 13 20 3 10 10 ...
 $ Consumer_profile: Factor w/ 2 levels "Business","Personal": 2 2 1 2 1 2 1 2 1 1 ...
 $ Product_category: Factor w/ 2 levels "Entertainment",..: 2 1 2 1 2 1 1 2 2 1 ...
 $ Product_type     : Factor w/ 2 levels "AC","TV": 1 2 1 2 1 2 2 1 1 2 ...
 $ AC_1001_Issue    : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 2 1 ...
 $ AC_1002_Issue    : Factor w/ 3 levels "0","1","2": 2 1 1 1 1 1 1 3 1 1 ...
 $ AC_1003_Issue    : Factor w/ 3 levels "0","1","2": 3 1 1 1 1 1 1 2 1 1 ...
 $ TV_2001_Issue    : Factor w/ 3 levels "0","1","2": 1 2 1 2 1 1 1 1 1 1 ...
 $ TV_2002_Issue    : Factor w/ 3 levels "0","1","2": 1 2 1 2 1 2 1 1 1 1 ...
 $ TV_2003_Issue    : Factor w/ 3 levels "0","1","2": 1 2 1 1 1 2 1 1 1 1 ...
 $ Claim_Value      : num  4474 25000 10000 4216 20000 ...
 $ Service_Centre   : Factor w/ 7 levels "10","11","12",..: 3 4 3 1 4 1 3 4 4 1 ...
 $ Product_Age      : int  202 60 3 672 3 275 10 7 6 4 ...
 $ Purchased_from   : Factor w/ 3 levels "Dealer","Internet",..: 3 1 1 1 3 1 3 3 1 1 ...
 $ Call_details     : num  30 1.3 2.5 25 6.5 11 1.6 1.6 1.4 0.5 ...
 $ Purpose          : Factor w/ 3 levels "Claim","Complaint",..: 1 2 1 3 1 1 1 1 2 2 ...
 $ Fraud            : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

PREPARATION

VISUALIZATION DATA FRAME (SOME)

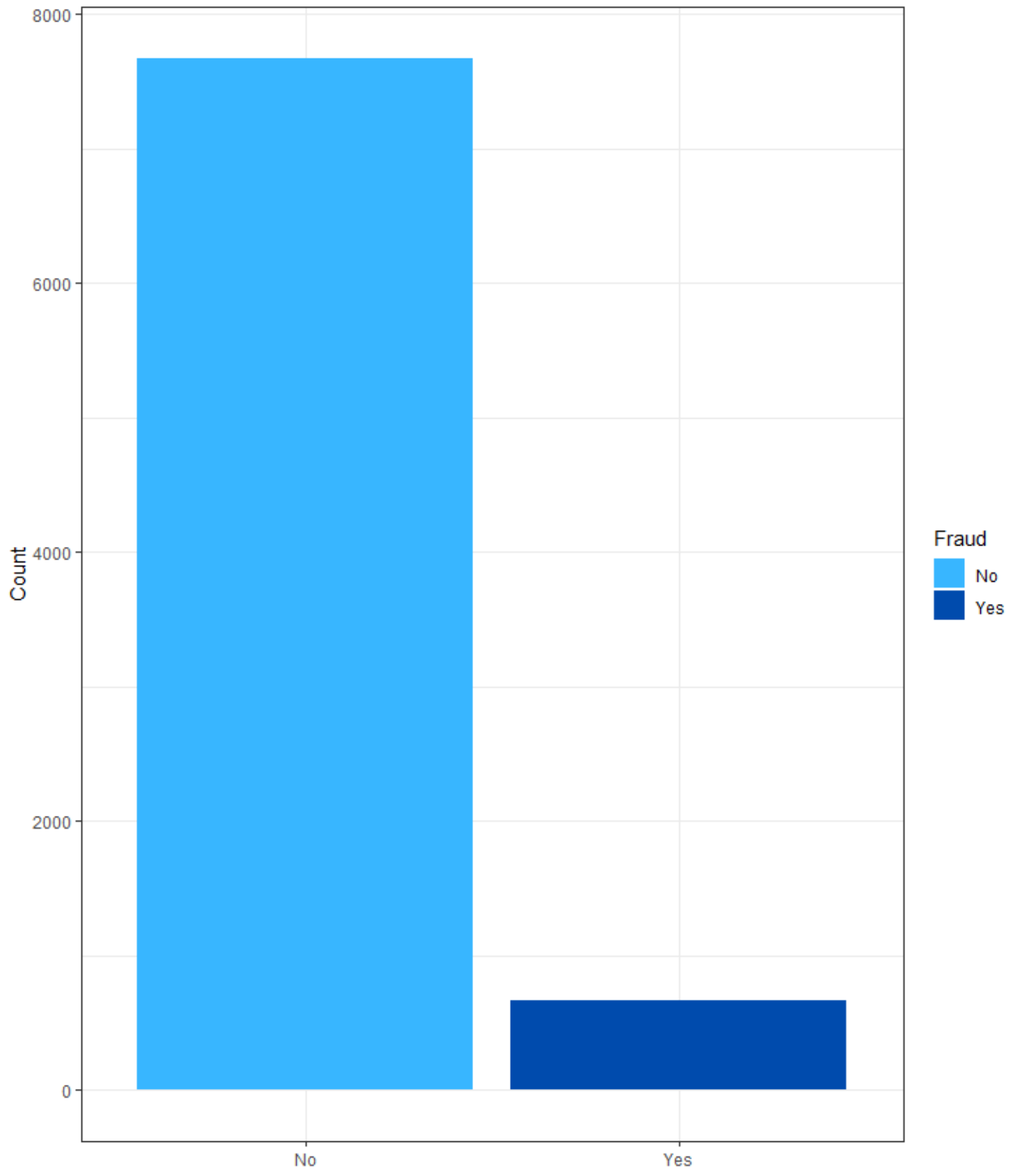
```
> str(df)
tibble [12,316 x 17] (S3: tbl_df/tbl/data.frame)
 $ Case_ID         : int [1:12316] 7957 7957 1396 1396 1396 5824 5824 6721 6721 3954 ...
 $ Region          : Factor w/ 8 levels "East","North",..: 3 3 1 1 1 8 8 3 3 1 ...
 $ State           : Factor w/ 22 levels "Andhra Pradesh",..: 2 2 4 4 4 6 6 22 22 11 ...
 $ Area            : Factor w/ 2 levels "Rural","Urban": 1 1 2 2 2 1 1 2 2 2 ...
 $ City            : Factor w/ 27 levels "Agartala","Ahmedabad",..: 9 9 17 17 17 2 2 13 13 3 ...
 $ Consumer_profile: Factor w/ 2 levels "Business","Personal": 2 2 2 2 2 2 2 2 2 2 ...
 $ Product_category: Factor w/ 2 levels "Entertainment",..: 2 2 1 1 1 1 1 1 1 2 ...
 $ Product_type    : Factor w/ 2 levels "AC","TV": 1 1 2 2 2 2 2 2 2 1 ...
 $ Claim_Value     : num [1:12316] 4474 4474 25000 25000 25000 ...
 $ Service_Centre  : Factor w/ 7 levels "10","11","12",..: 3 3 4 4 4 1 1 1 1 4 ...
 $ Product_Age     : int [1:12316] 202 202 60 60 60 672 672 275 275 7 ...
 $ Purchased_from  : Factor w/ 3 levels "Dealer","Internet",..: 3 3 1 1 1 1 1 1 1 3 ...
 $ Call_details    : num [1:12316] 30 30 1.3 1.3 1.3 25 25 11 11 1.6 ...
 $ Purpose         : Factor w/ 3 levels "Claim","Complaint",..: 1 1 2 2 2 3 3 1 1 1 ...
 $ Fraud           : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ Issue           : chr [1:12316] "AC_1002_Issue" "AC_1003_Issue" "TV_2001_Issue" "TV_2002_Issue" ...
 $ Issue_type      : Factor w/ 3 levels "0","1","2": 2 3 2 2 2 2 2 2 2 3 ...
```

Data exploration & Visualization

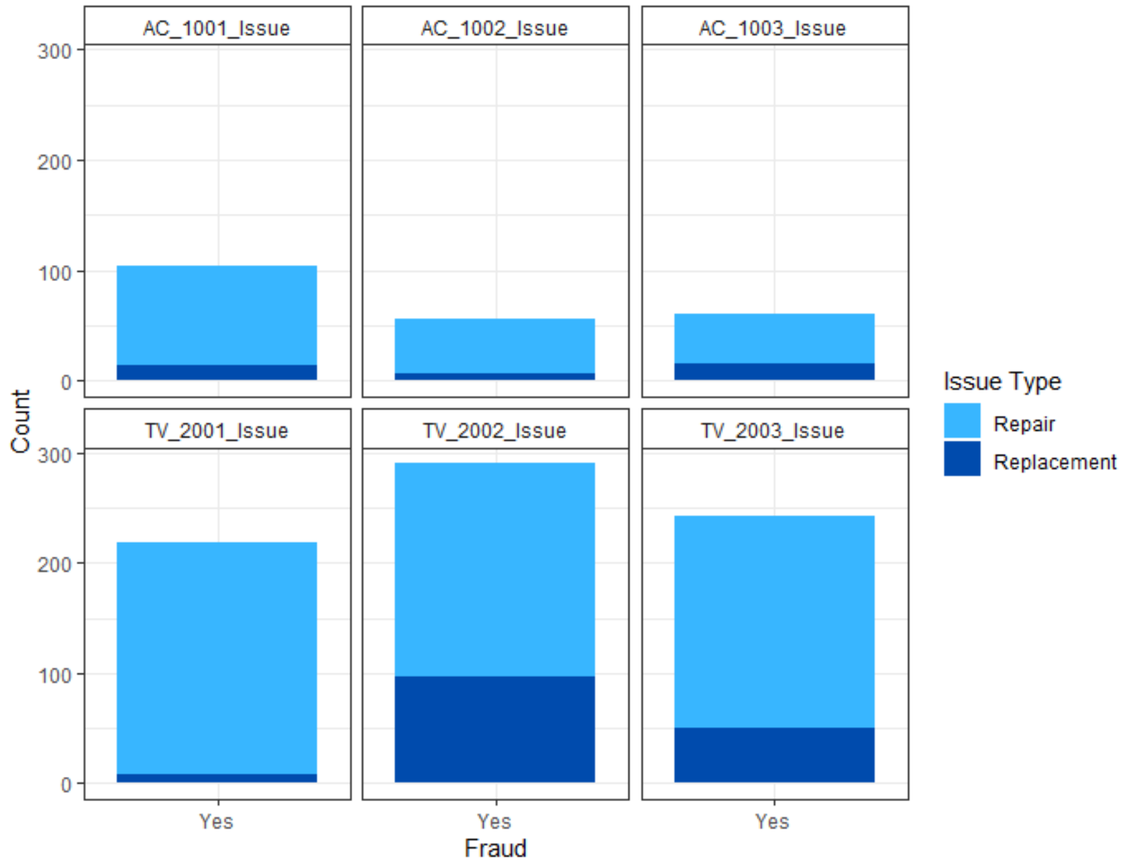


31 graphs

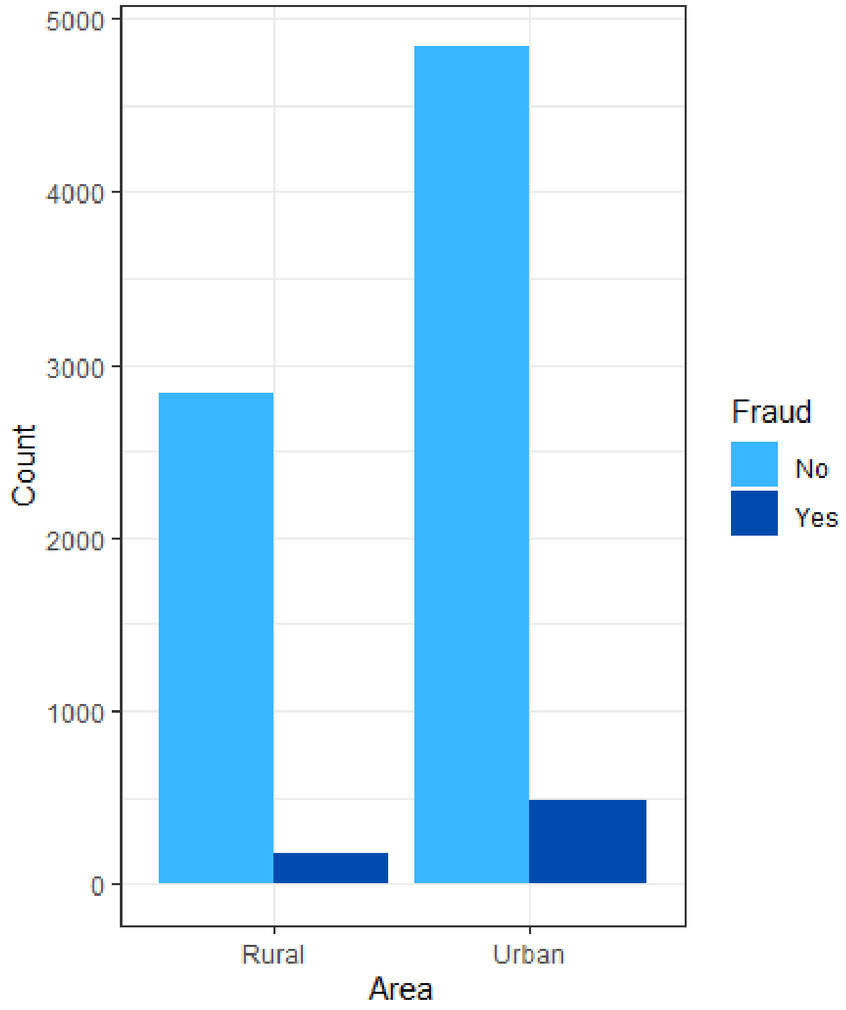
Fraud per Count



Issue type given that Fraud of each issue category

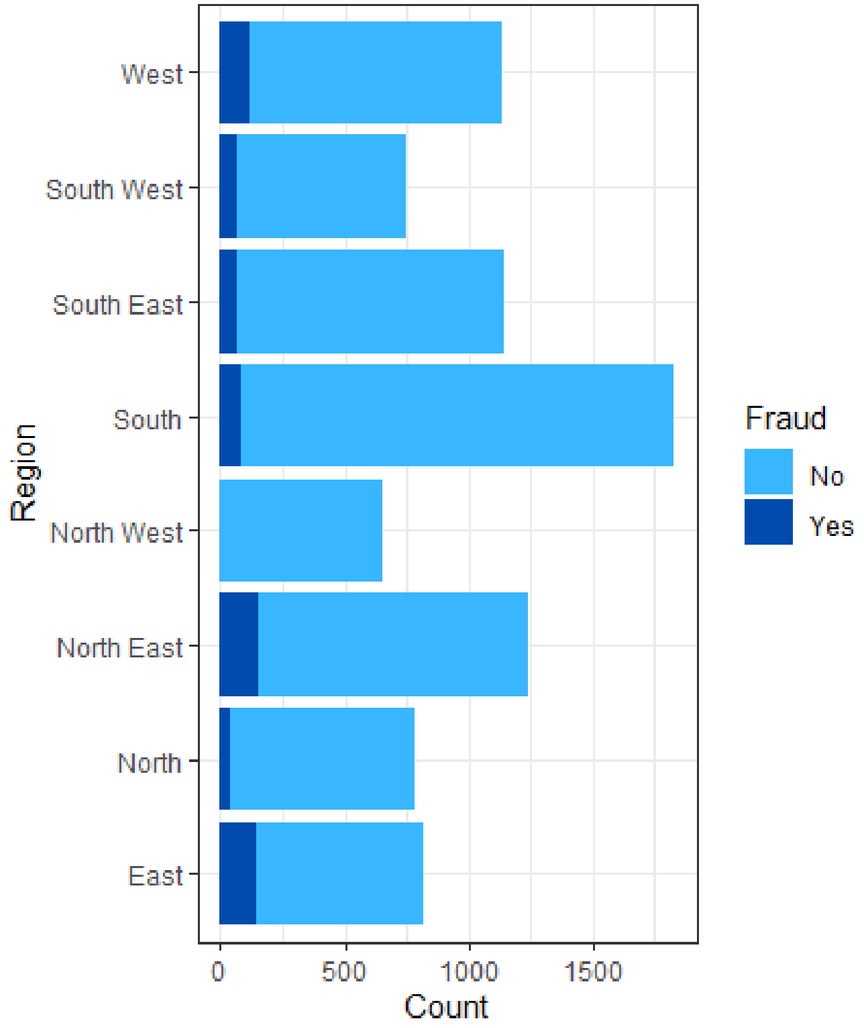


Fraud count per Area

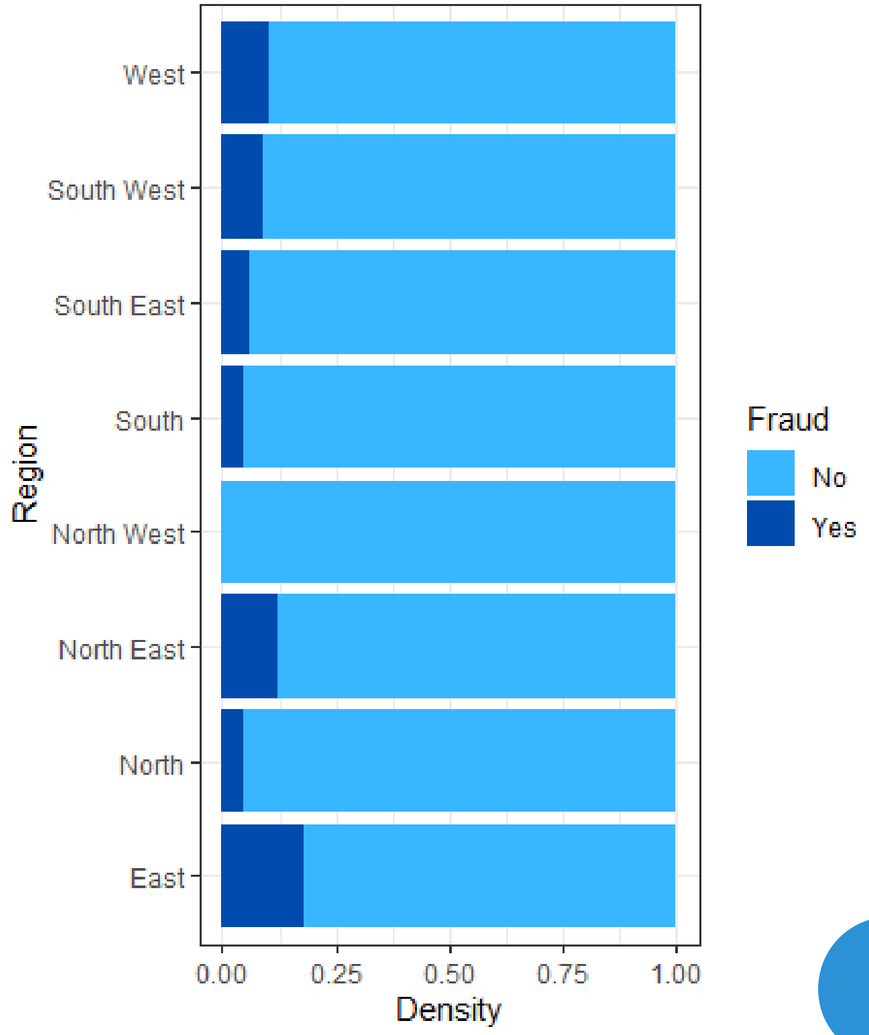


Fraud density per Area

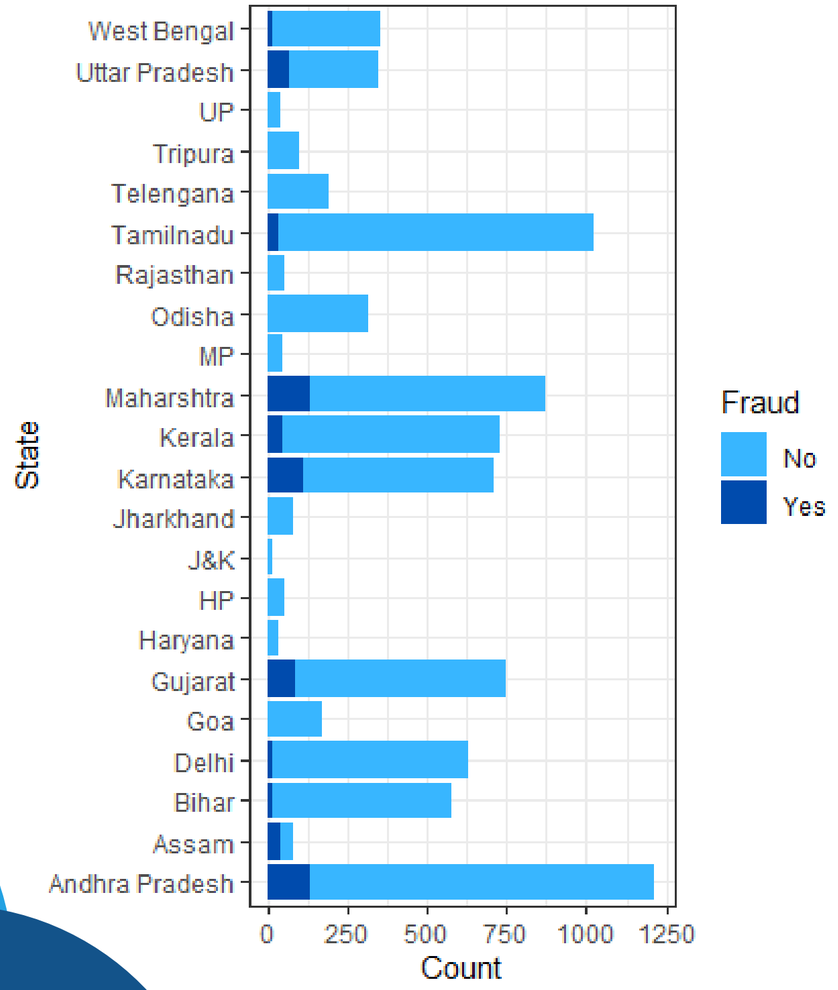
Fraud count in each Region



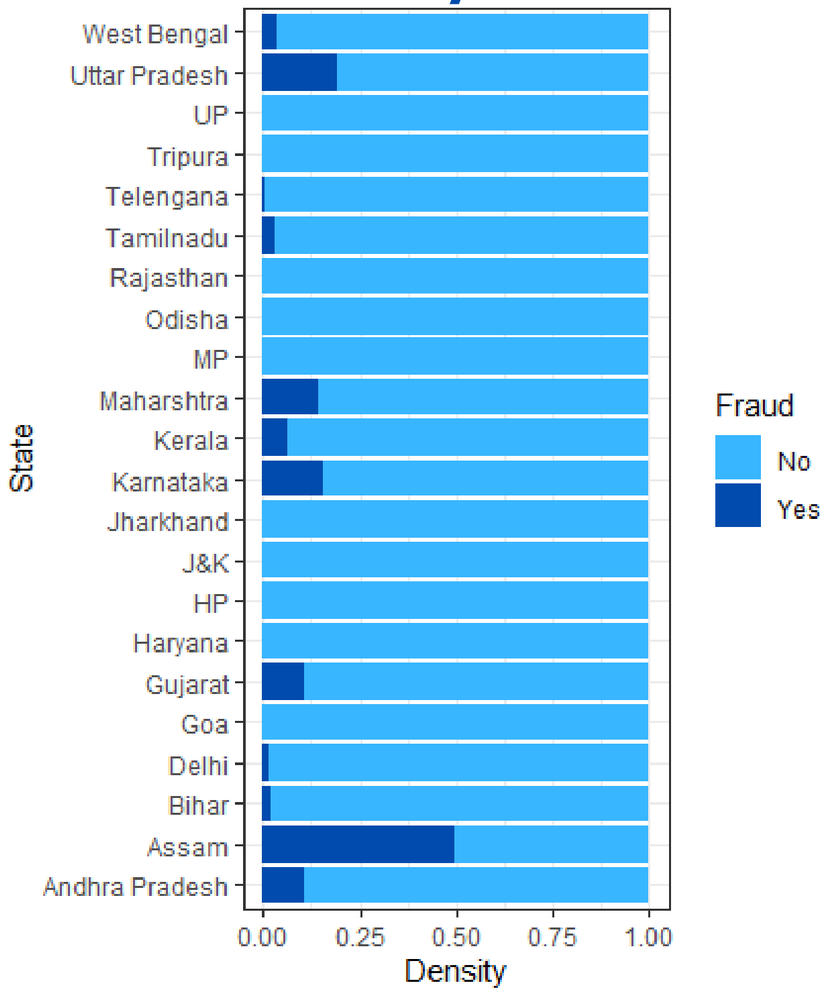
Fraud density in each Region



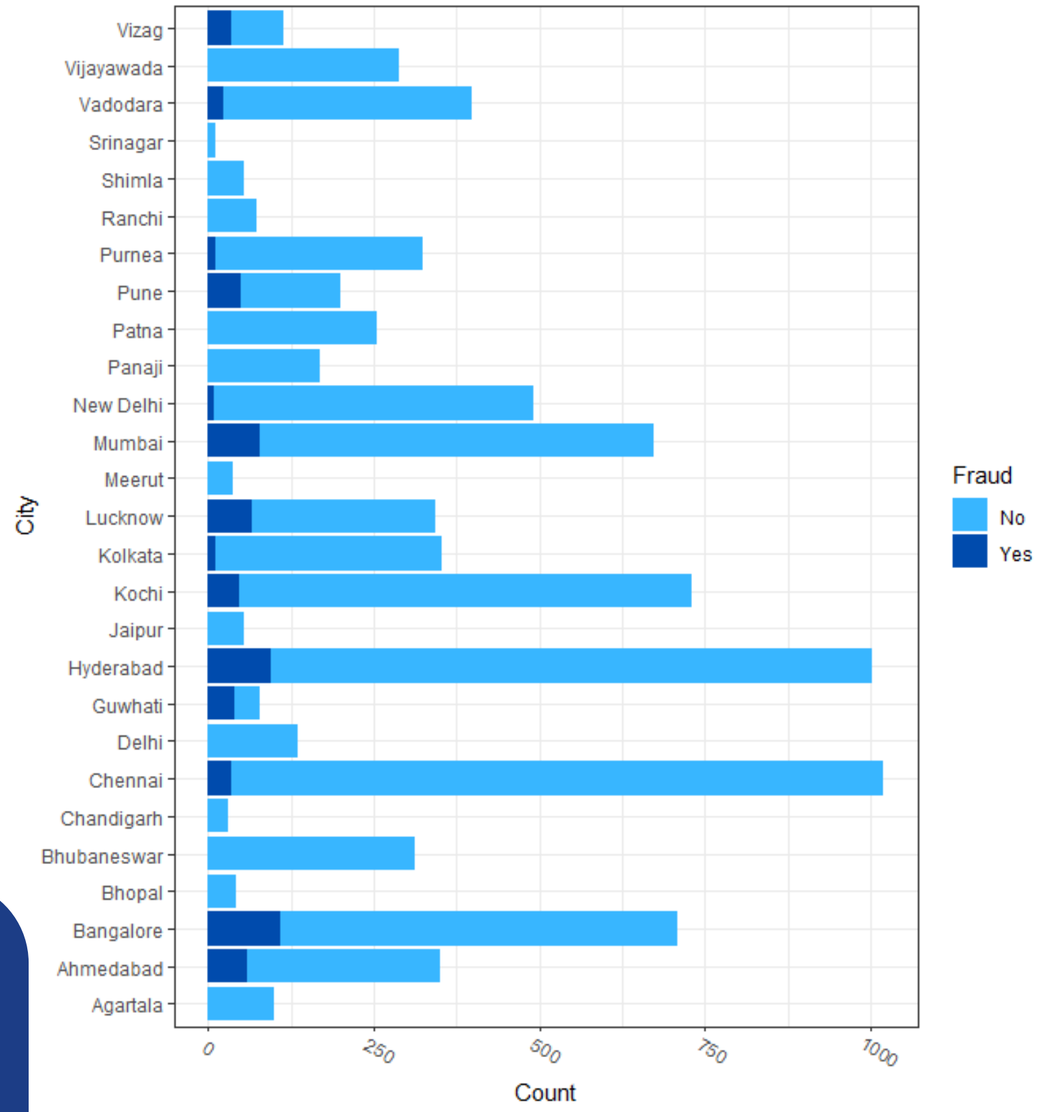
Fraud count in each State



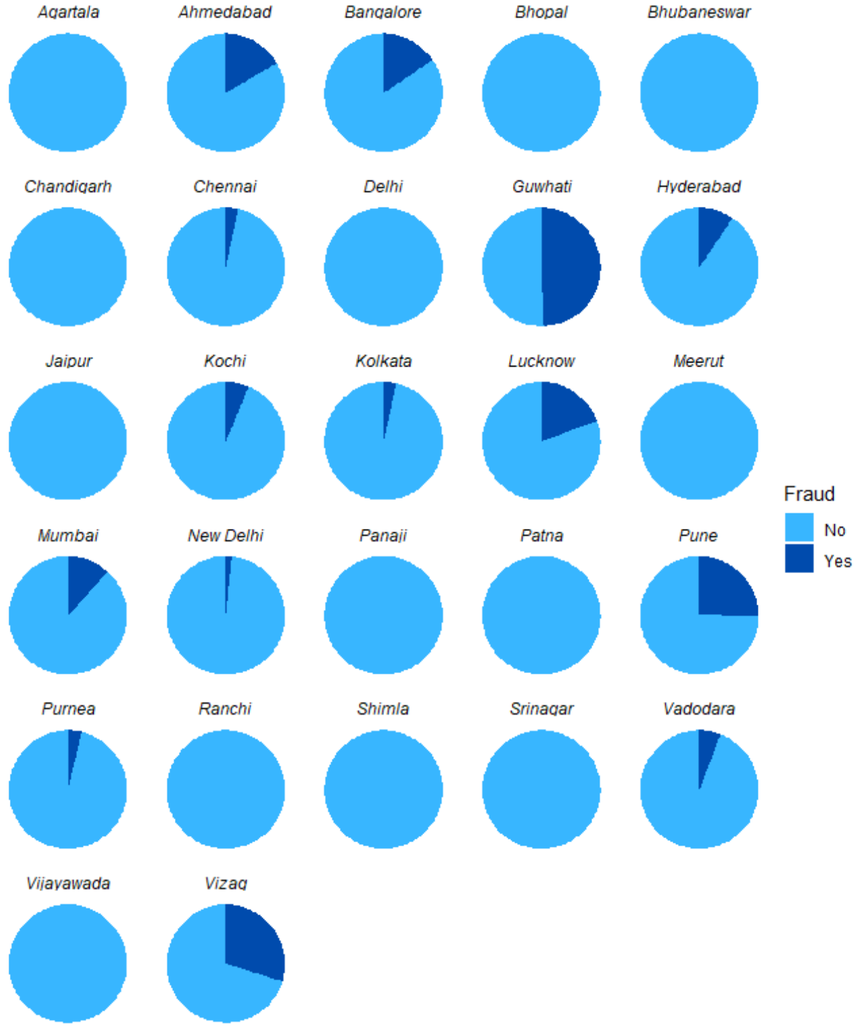
Fraud density in each State



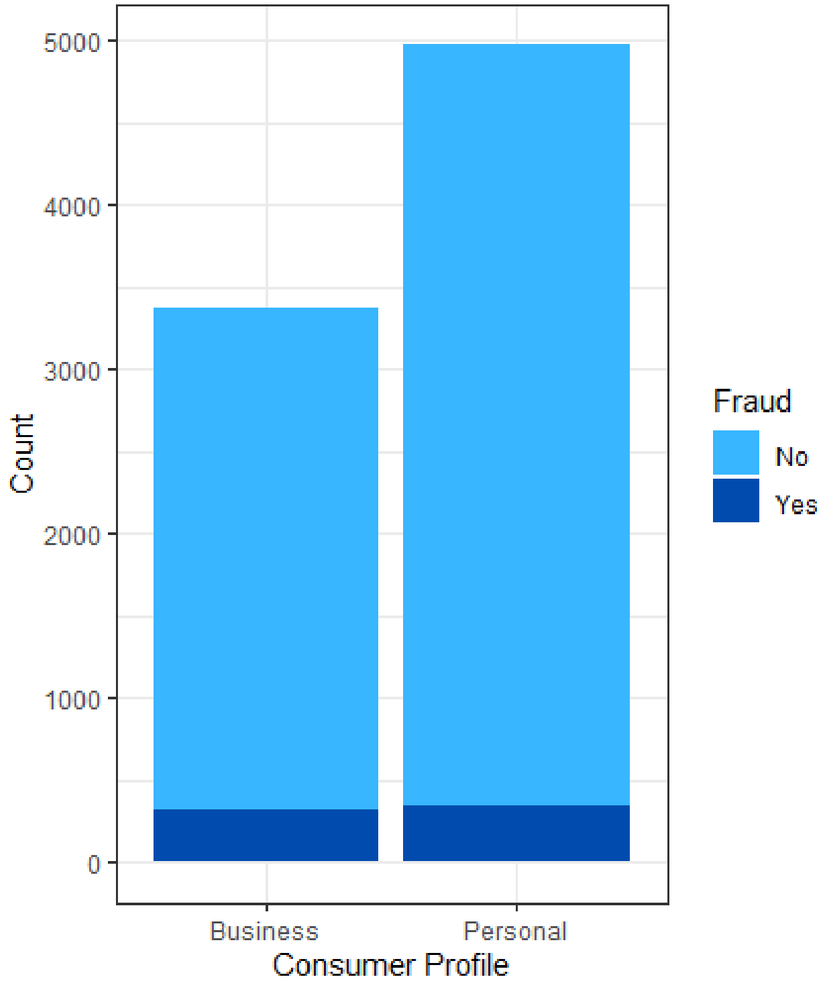
Fraud count in each City



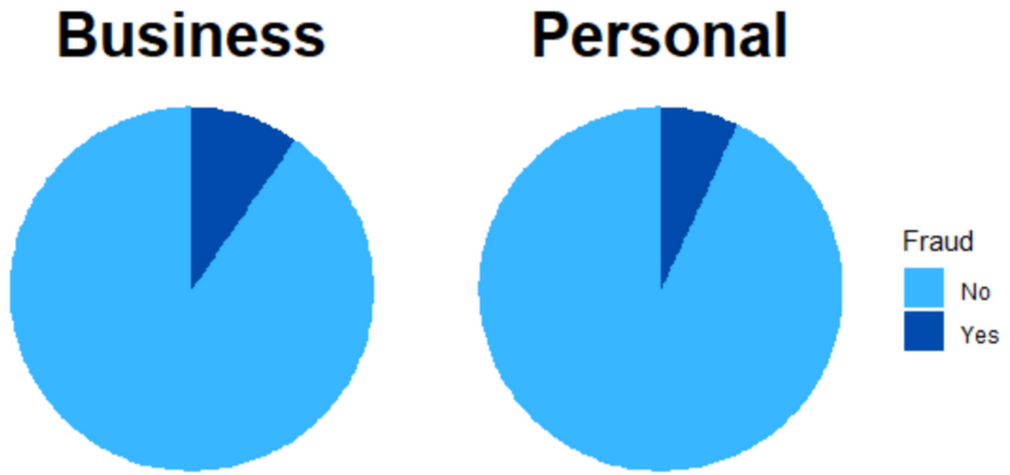
Fraud density in each City



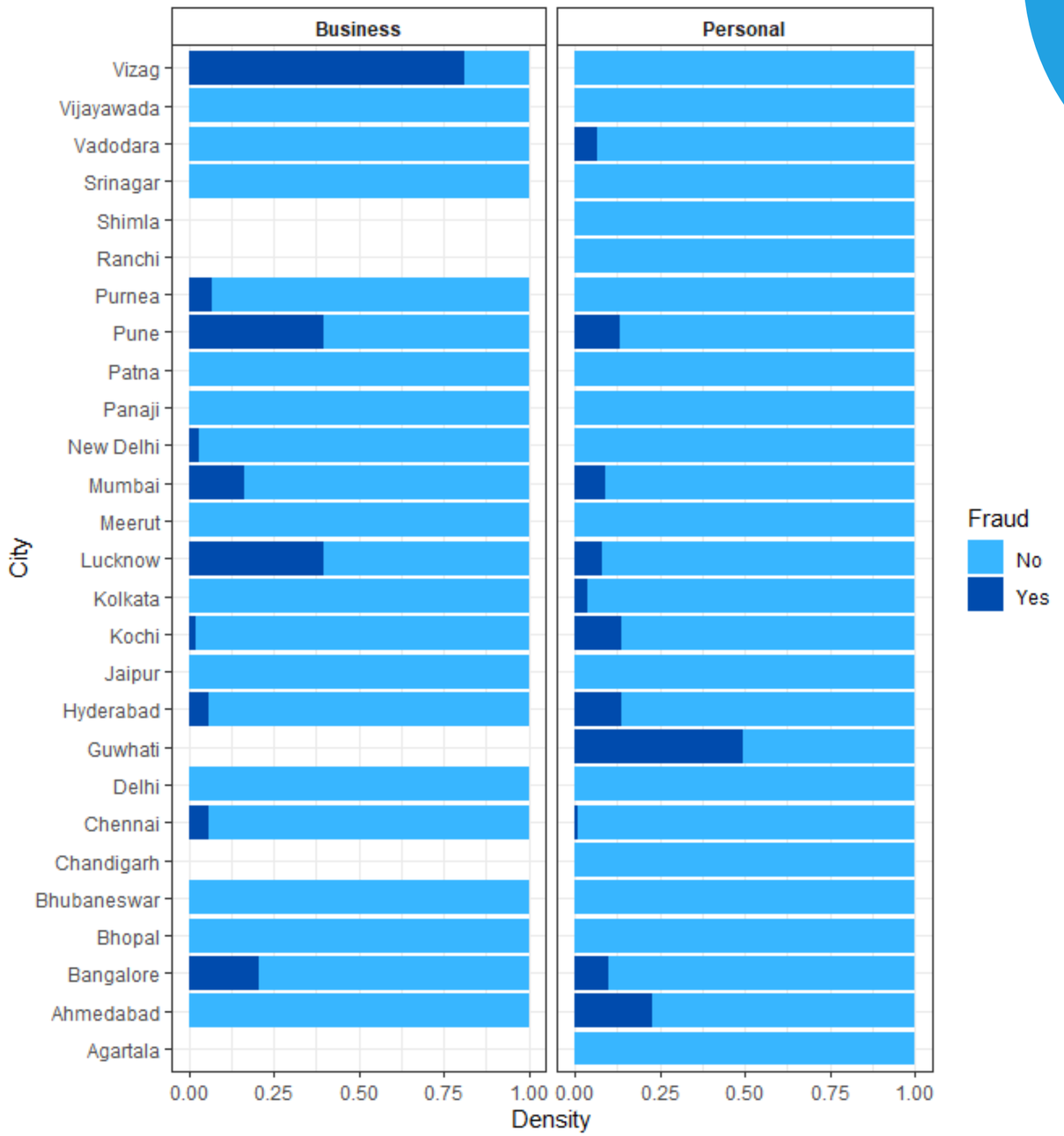
Fraud count in each Consumer Profile



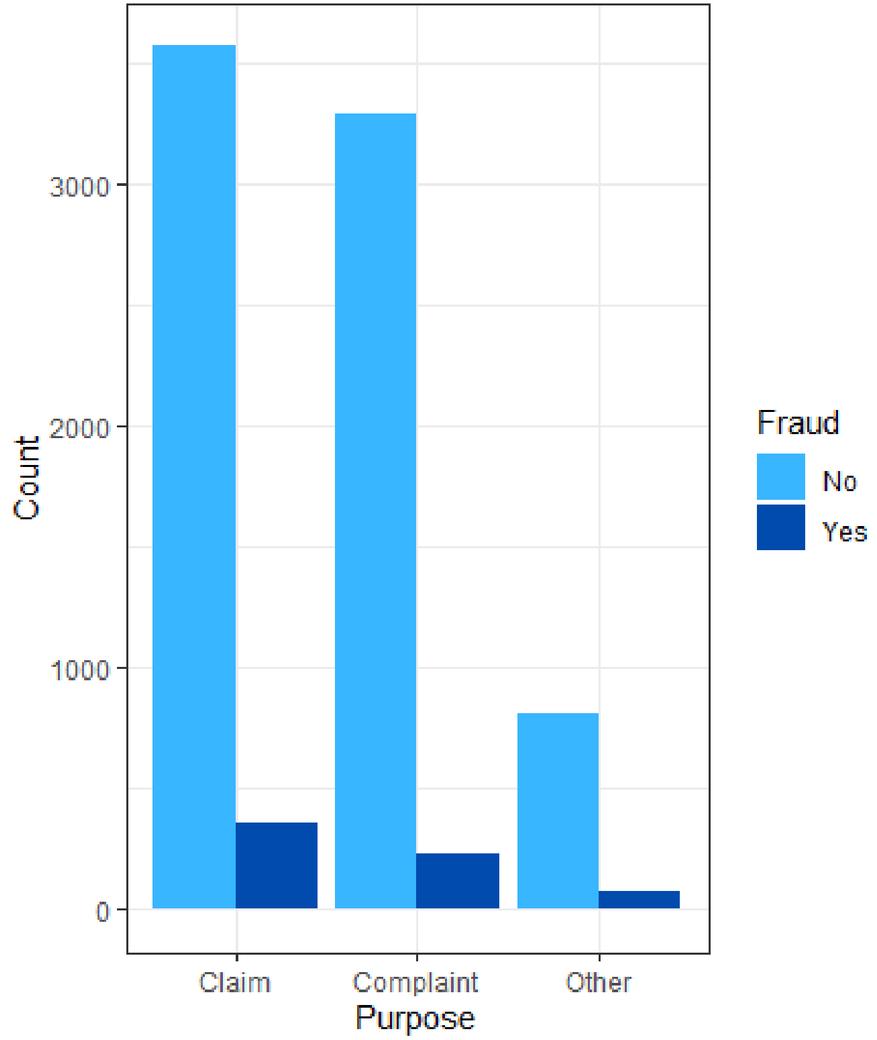
Fraud density in each Consumer Profile



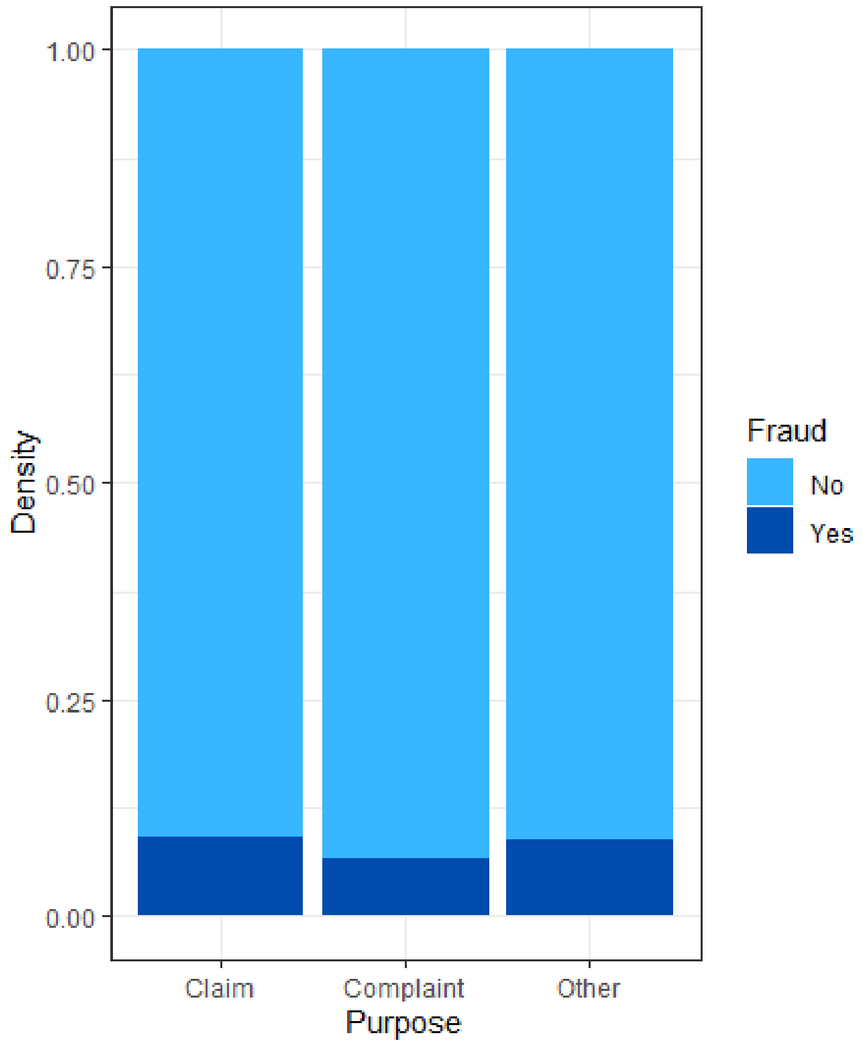
Fraud density in each Consumer Profile

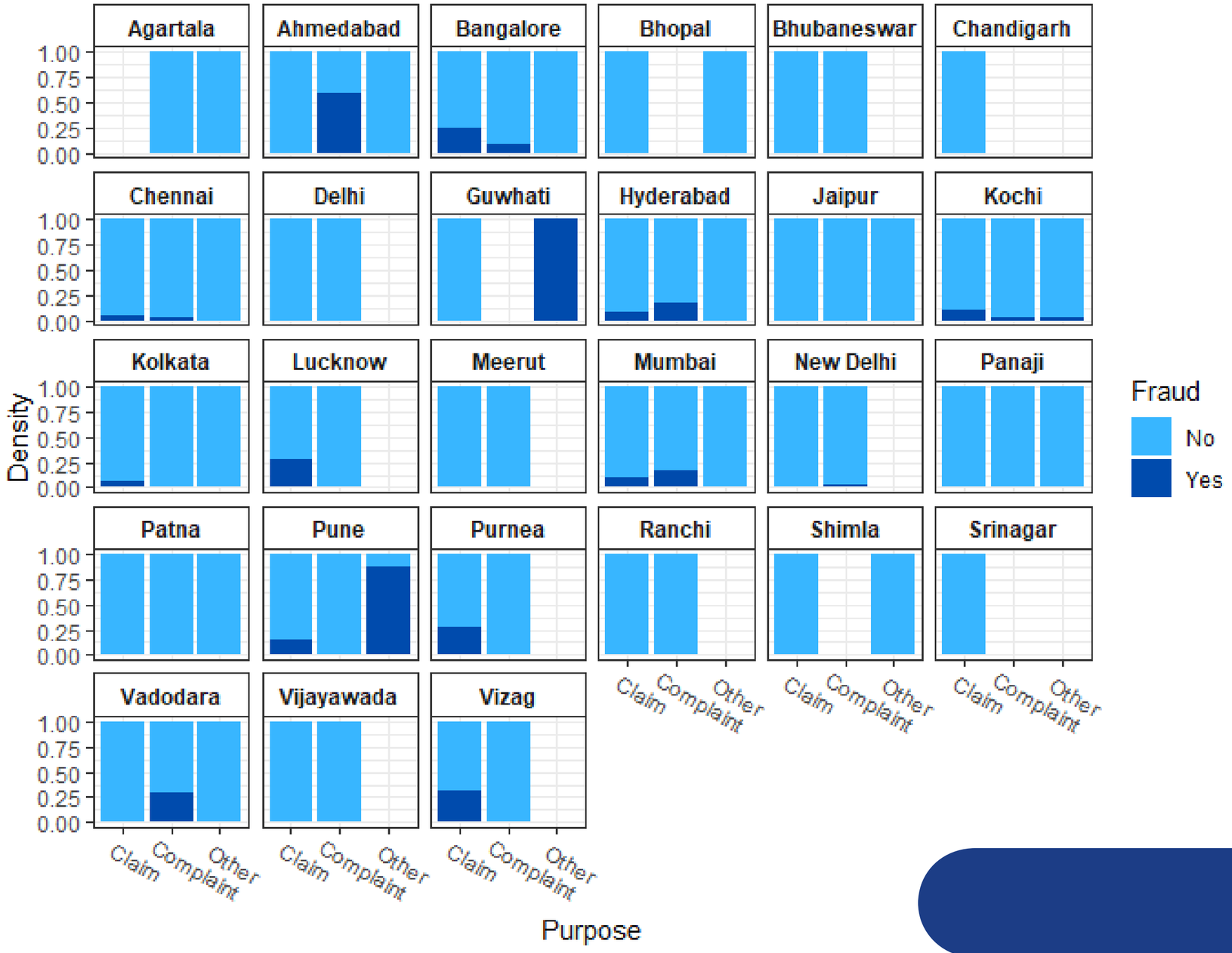


Fraud count in each Purpose

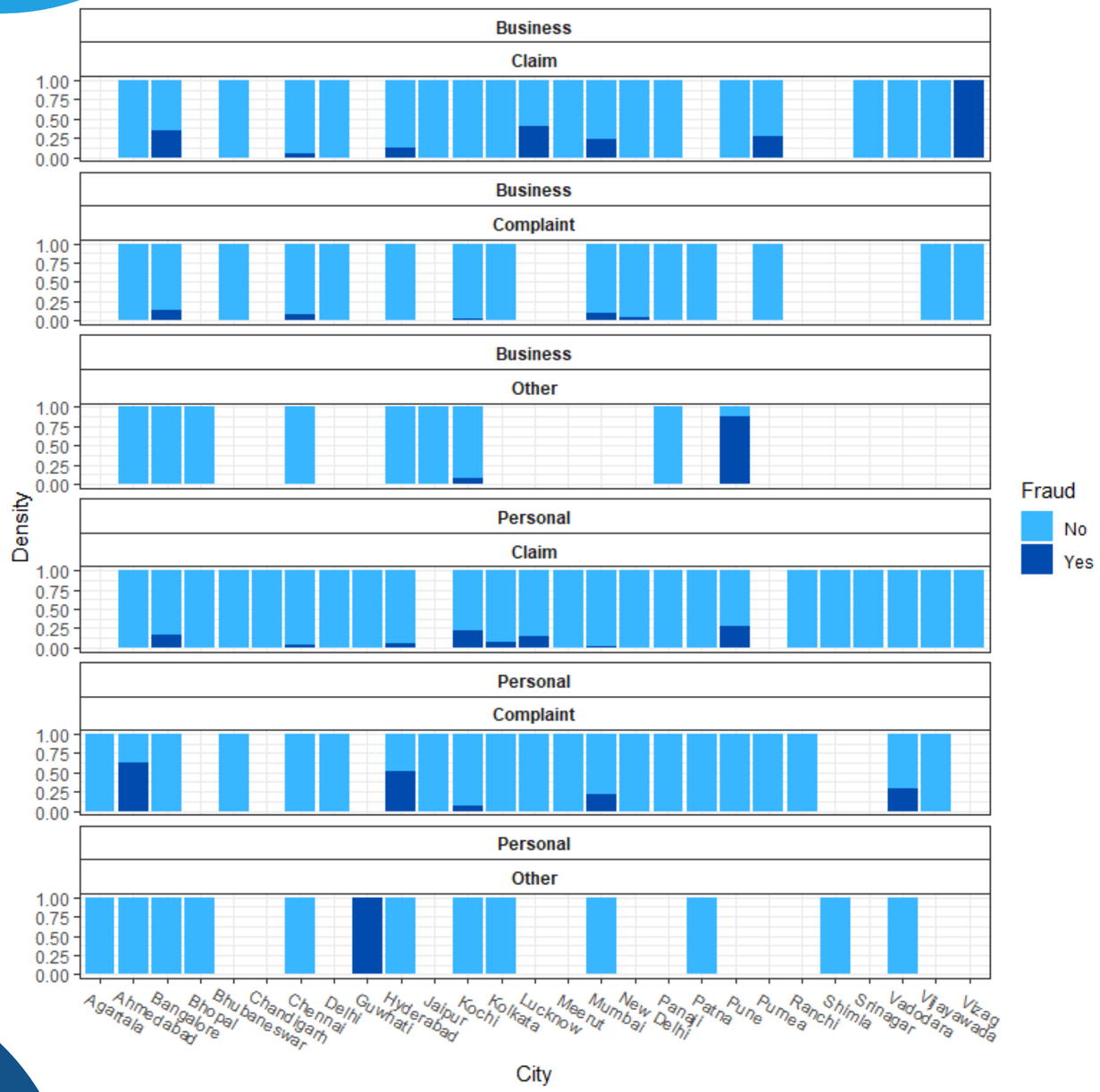


Fraud density in each Purpose



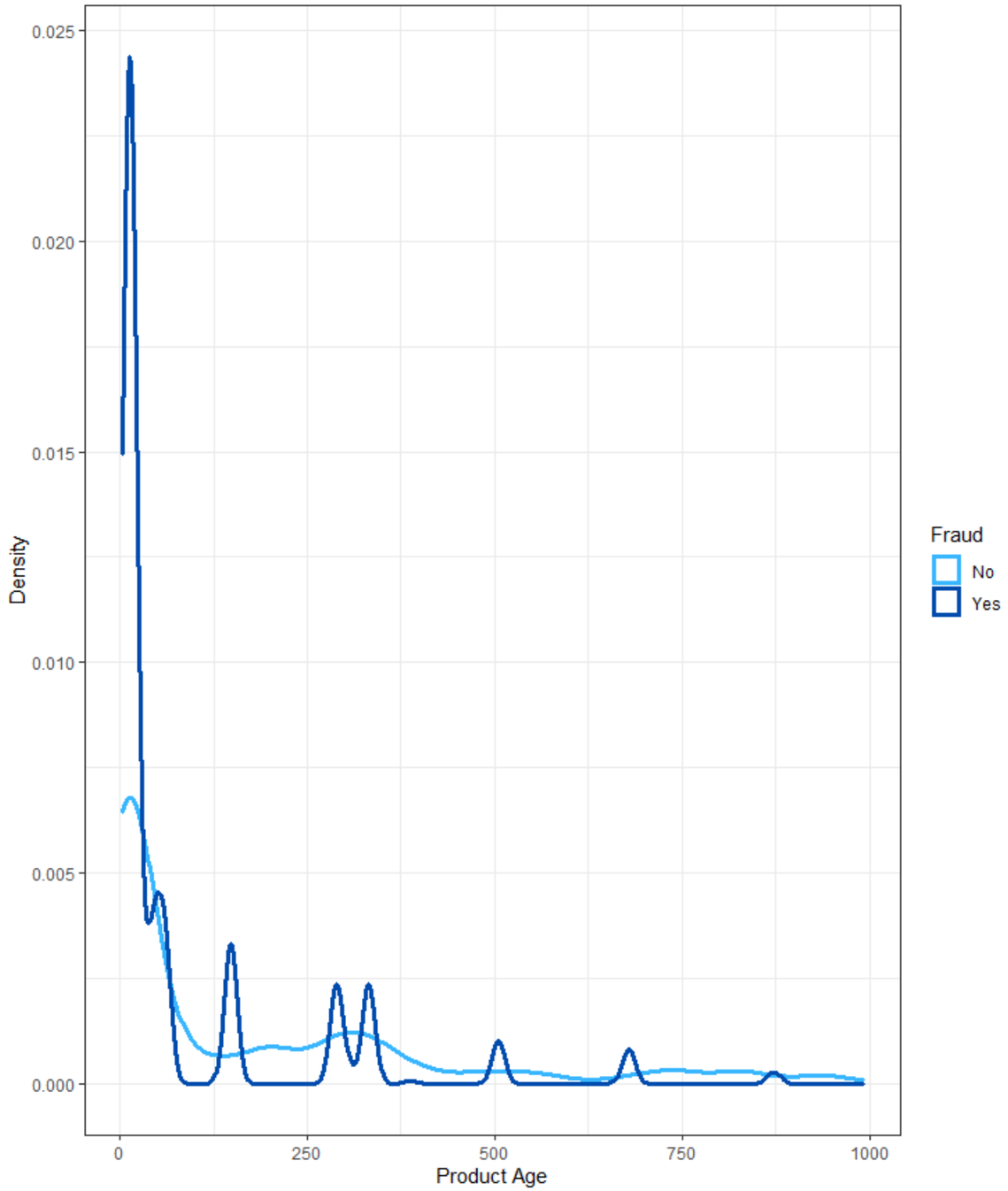


Fraud density in each Purpose from different City

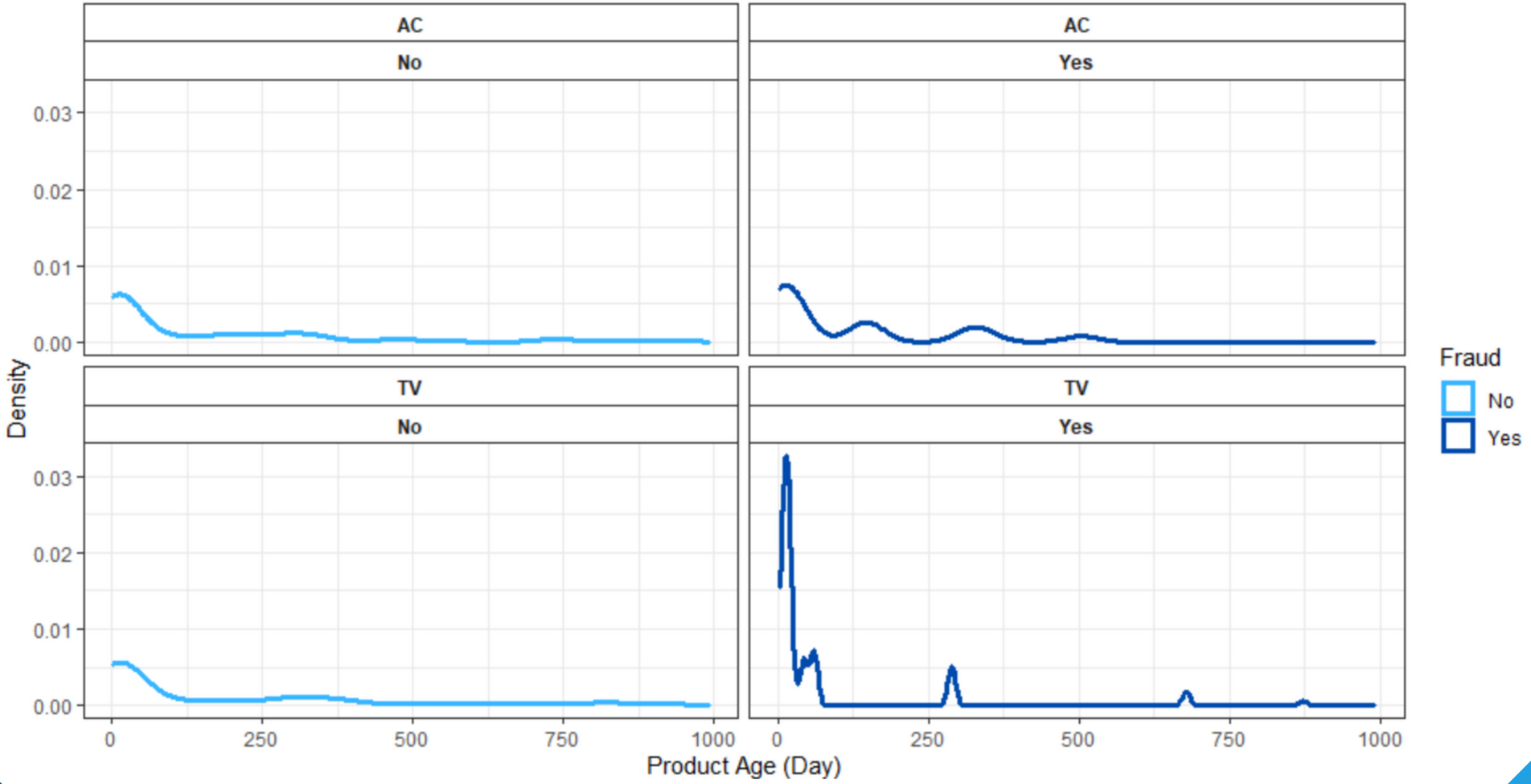


Fraud density by Purpose and Consumer profile from different City

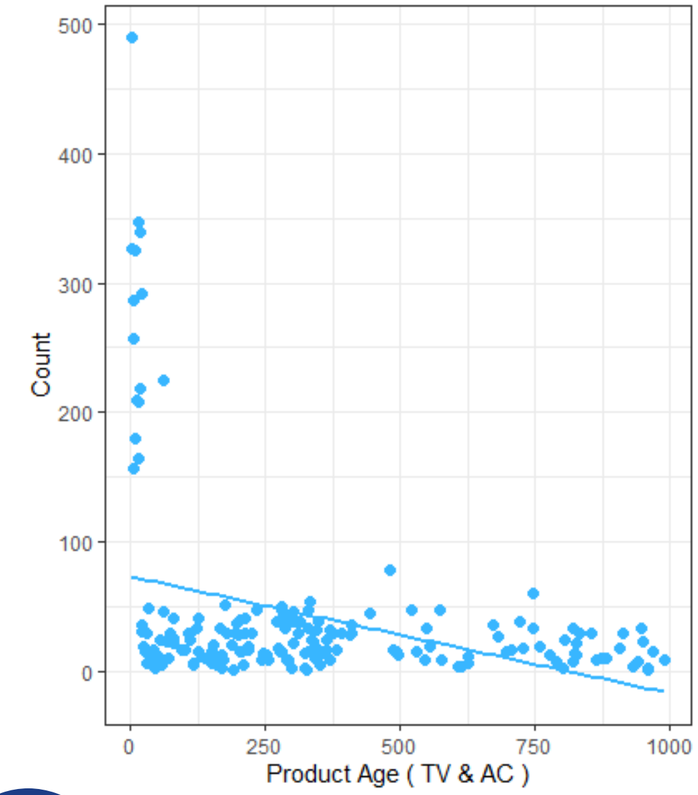
Fraud density by Product age



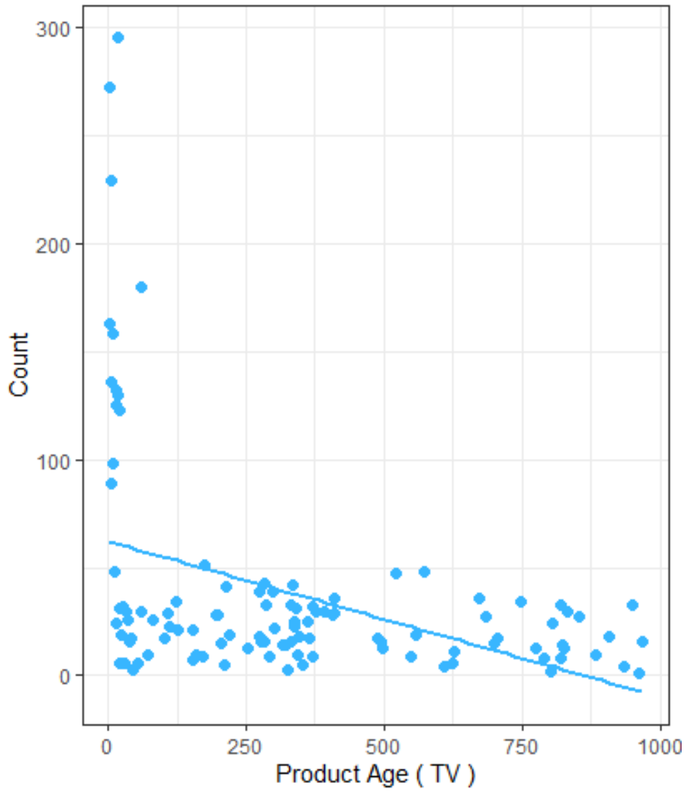
Fraud density per Product age from each type of product



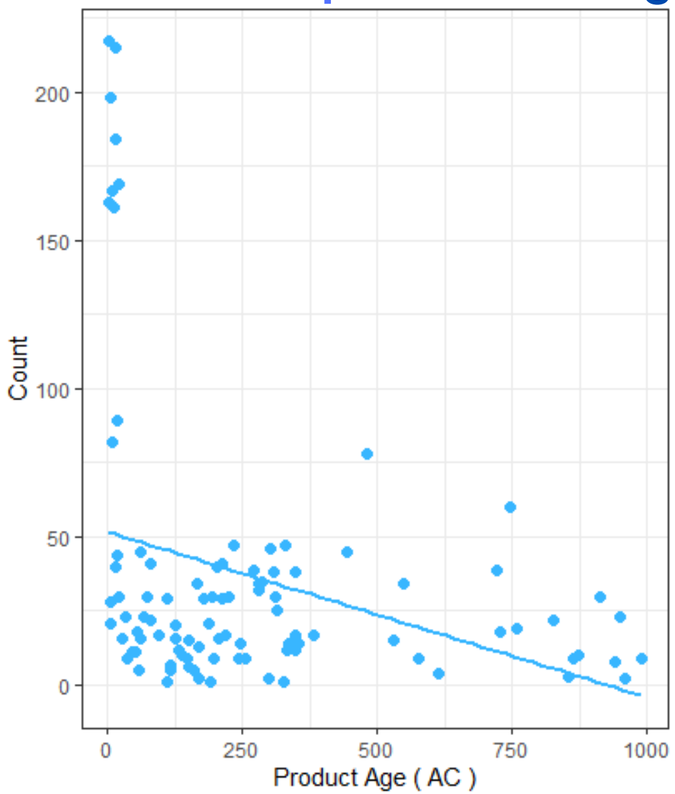
Count of TV&AC per Product age



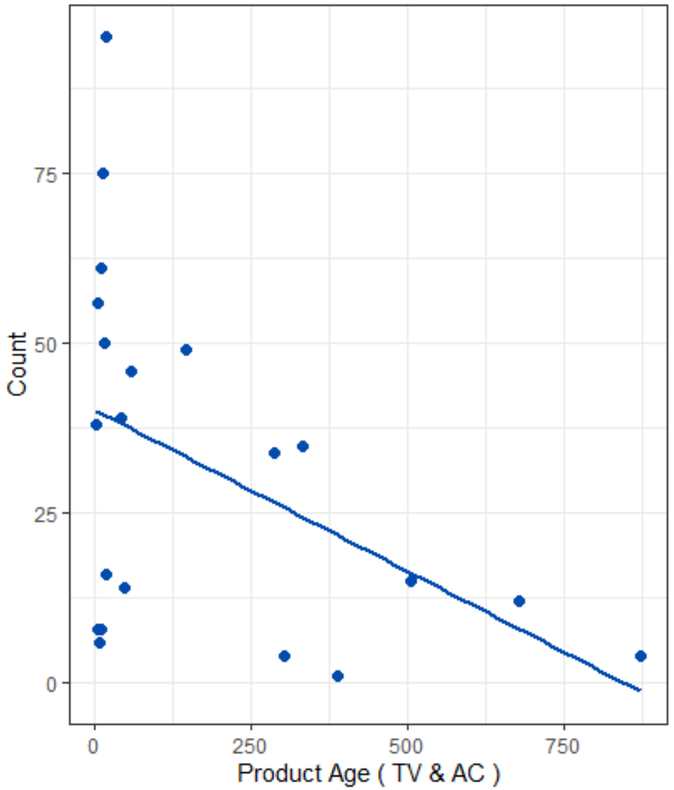
Count of TV per Product age

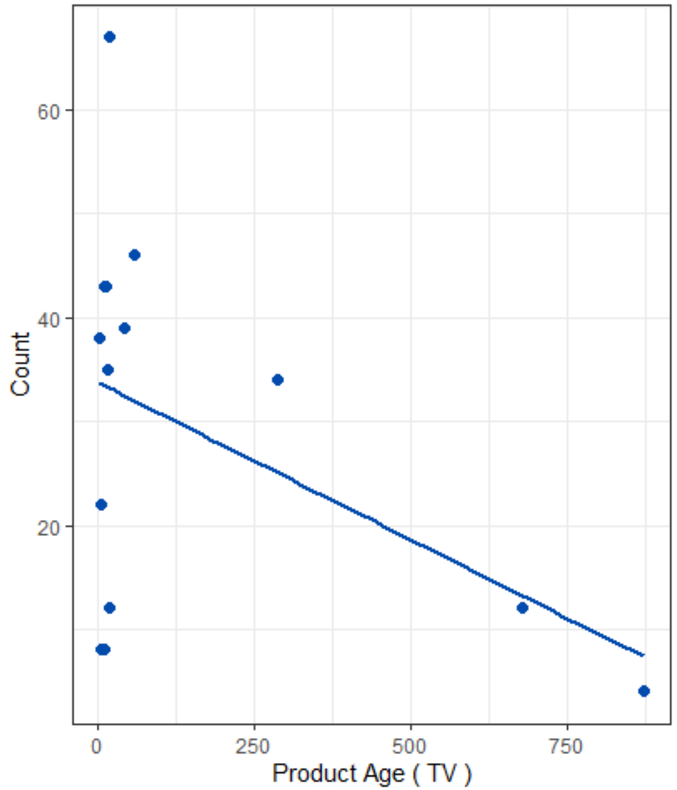


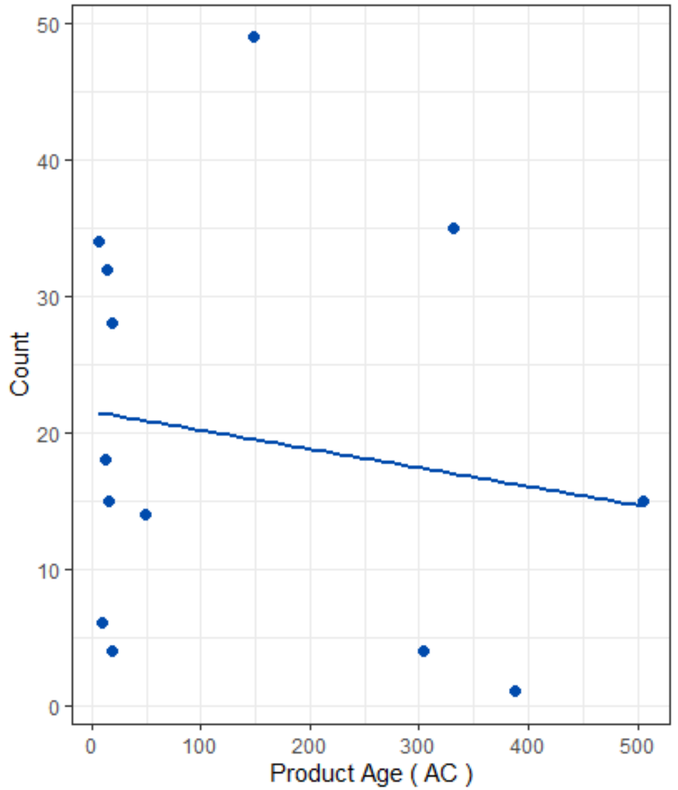
Count of AC per Product age



Fraud count from TV per Product age



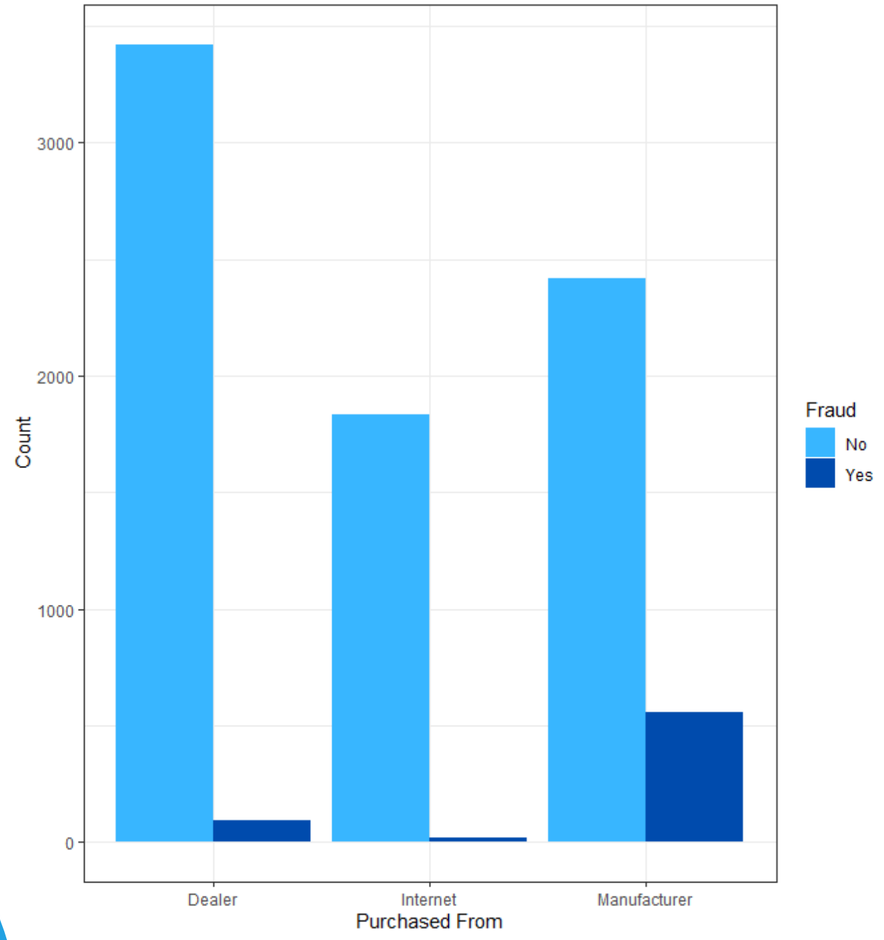




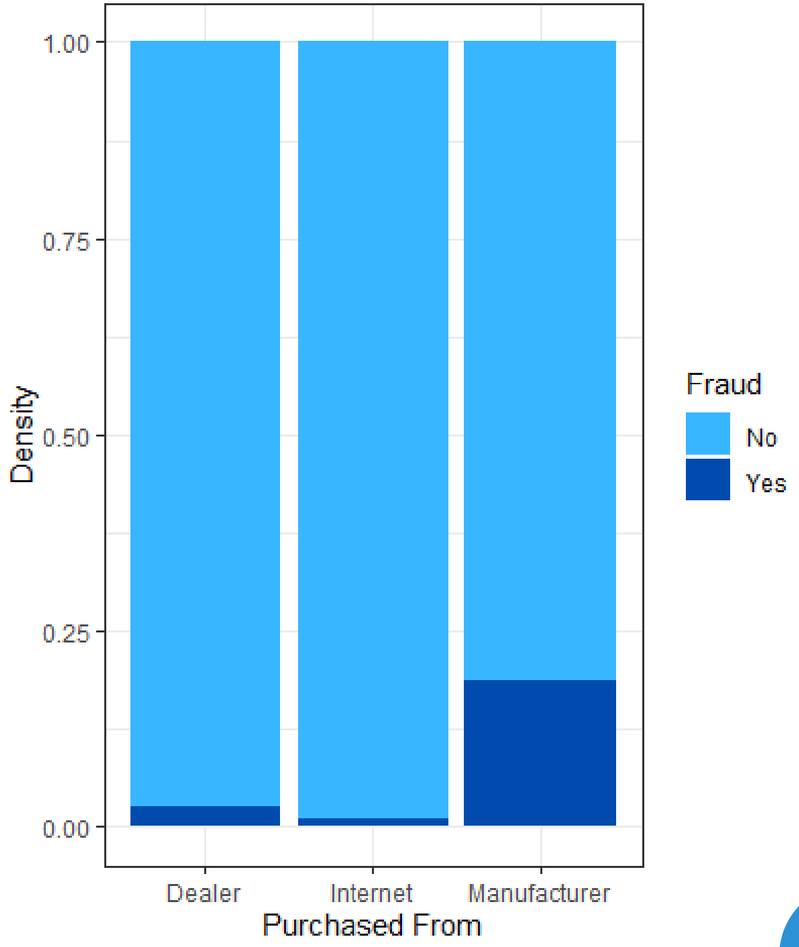
Fraud count from TV&AC per Product age

Fraud count from AC per Product age

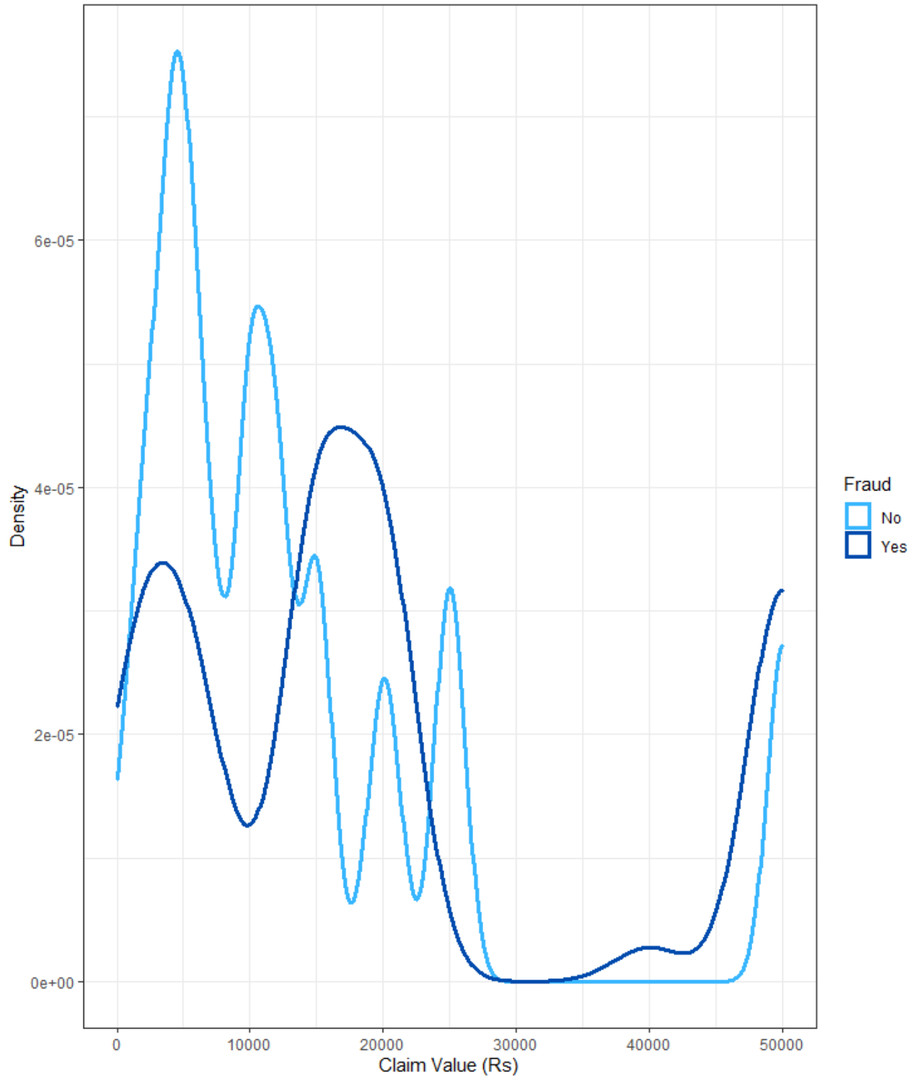
Fraud count per which way the consumer purchased from



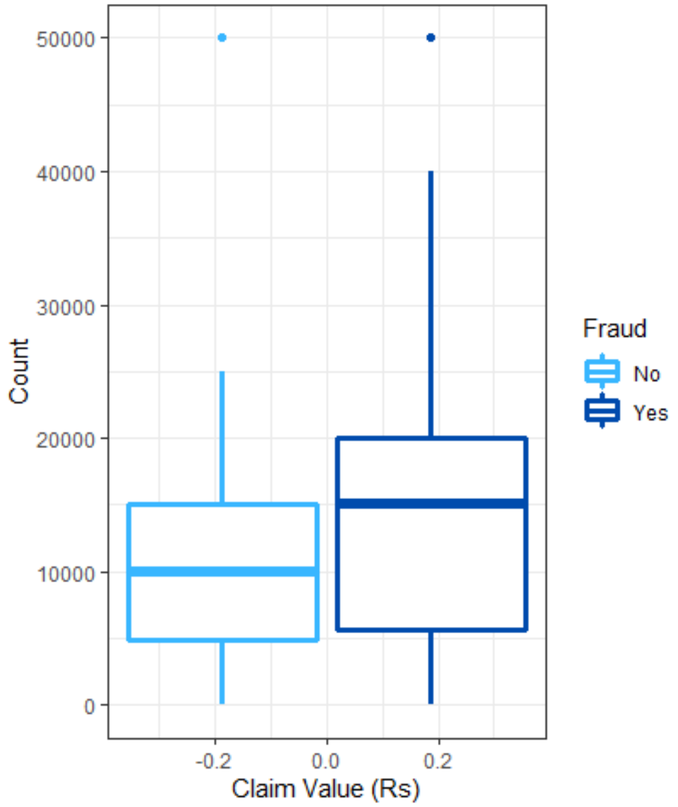
Fraud density per which way the consumer purchased from



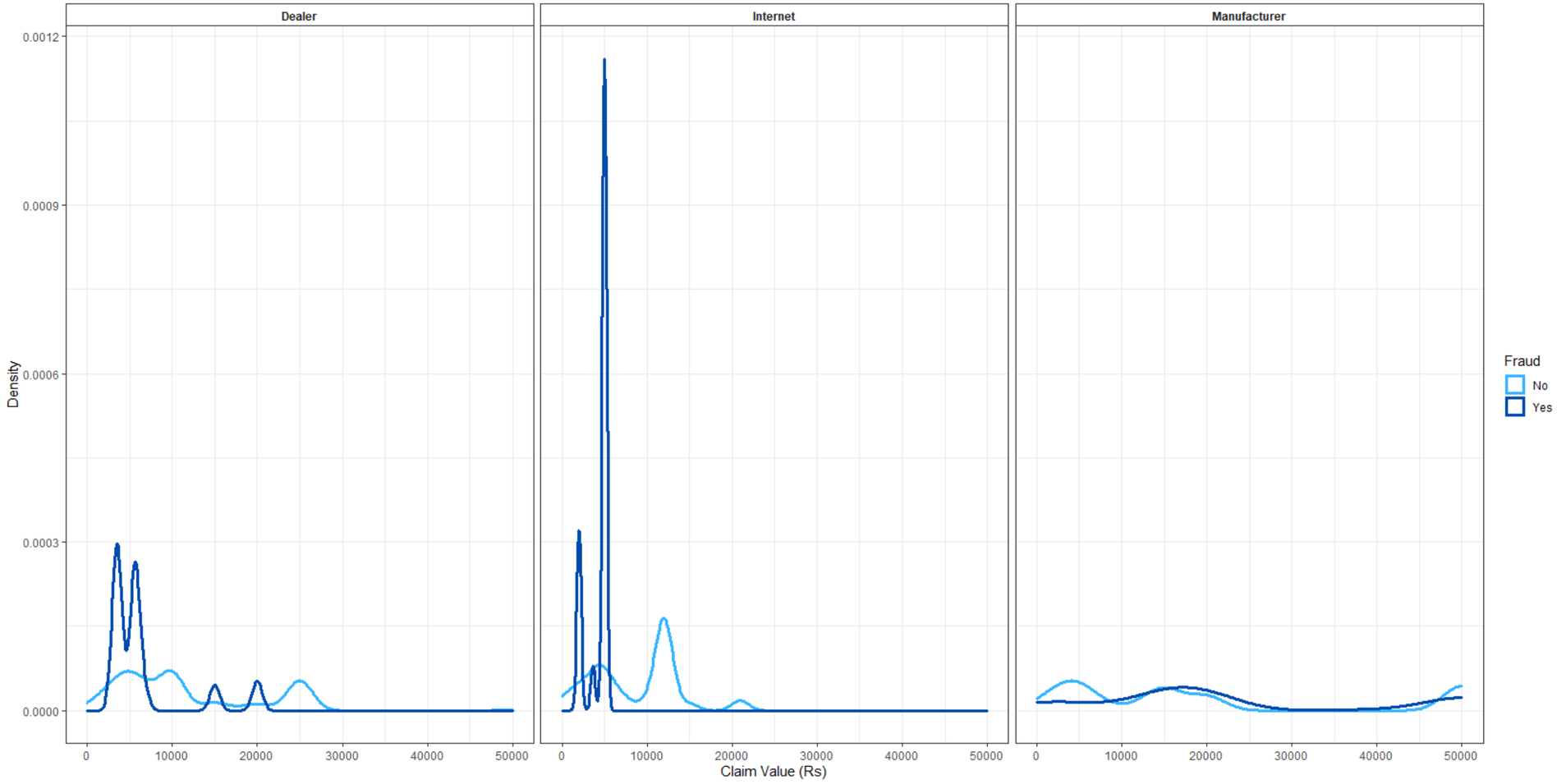
Fraud density compare with Claim Value



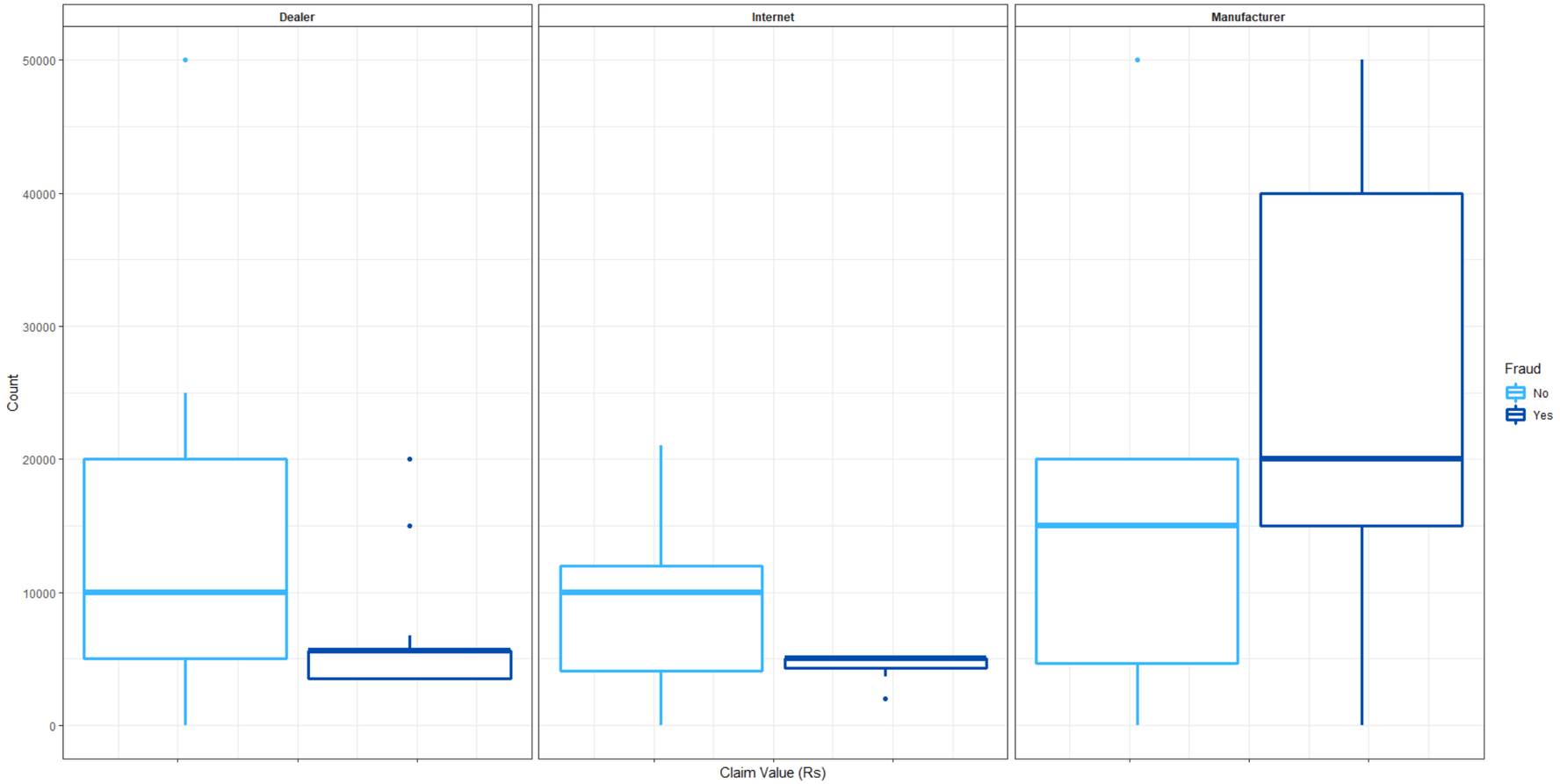
Fraud count per Claim Value



Fraud density per which way the consumer purchased from



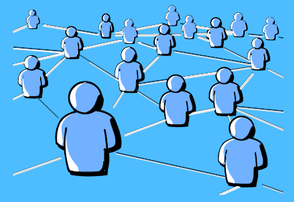
Fraud count per Claim Value separate in which way consumer Purchased from



Model explanation

How many model are there?

Relationship between variable and formula



LOGISTIC REGRESSION

DECISION TREE

Sensitivity = as high as possible

False negative = as low as possible

To detect most frauder

The reason

Necessity loss of money

Trust in company

Implementation

```r
model <- glm(Fraud ~. , data = train_set, family = binomial)

model <- glm(Fraud ~ City * Area + State + Region +
               Consumer_profile * City +
               Consumer_profile * Product_Age * Product_type +
               Purchased_from, data = train_set, family = binomial)

model <- glm(Fraud ~ City * Area + State + Region +
               Consumer_profile + Product_Age * Product_type +
               Purchased_from , data = train_set, family = binomial)

model <- glm(Fraud ~ Consumer_profile * City * Area + State + Region +
               Product_Age * Product_type +
               Purchased_from, data = train_set, family = binomial)

model <- glm(Fraud ~ Consumer_profile * City * Purpose +
               Area + State + Region +
               Product_Age * Product_type + Purchased_from +
               TV_2001_Issue * TV_2002_Issue * TV_2003_Issue +
               AC_1001_Issue * AC_1002_Issue * AC_1003_Issue,
             data = train_set, family = binomial)
```

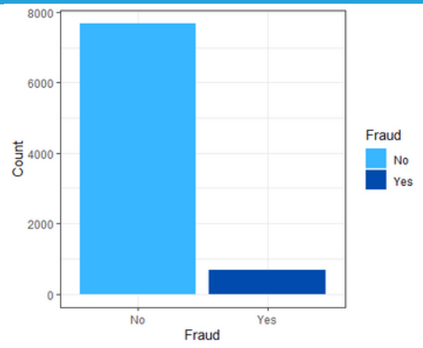
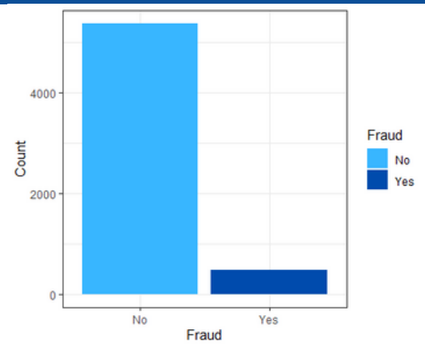
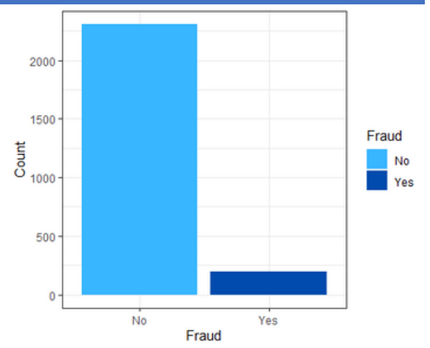
```
model <- glm(Fraud ~ Consumer_profile * City * Purpose + Area + State + Region +
               Product_Age * Product_type + Service_Centre +
               Purchased_from * Claim_Value  +
               TV_2001_Issue * TV_2002_Issue * TV_2003_Issue +
               AC_1001_Issue * AC_1002_Issue * AC_1003_Issue,
               data = train_set, family = binomial )
```

HIGHEST TRUE POSITIVE

```
tree <- rpart(Fraud ~ Product_Age + Claim_Value + Purchased_from +
              State + Region + Call_details + Service_Centre,
              data = train_set, control = rpart.control(cp = 0.001))
```

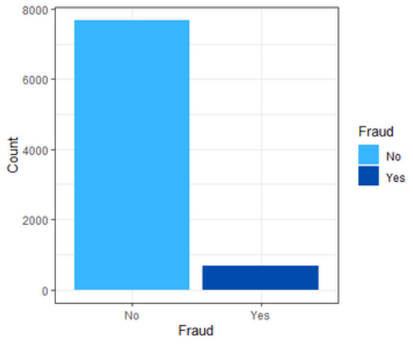
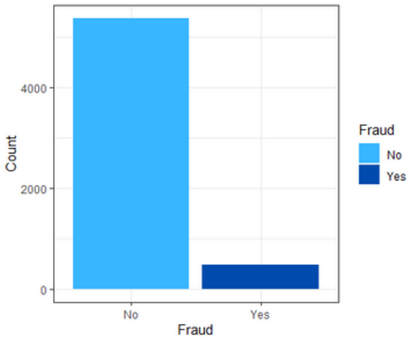
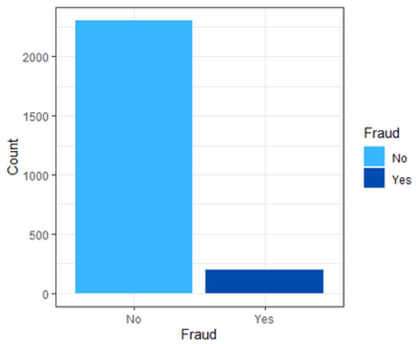
HIGHEST KAPPA

```
tree <- rpart(Fraud ~ Product_Age + Claim_Value + Purchased_from +
              City + Purpose + Region + Call_details + Service_Centre,
              data = train_set, control = rpart.control(cp = 0.001))
```

FULL	TRAIN	TEST
		
Fraud Percent(Yes): 7.98%	8.06%	7.79%


```
set.seed(123)
test_index <- sample(nrow(claims_df), 0.3*nrow(claims_df))
train_set <- claims_df[-test_index,]
test_set <- claims_df[test_index,]
```

Test Set contain 30% of Data
Train Set contain 70% of Data

		
Fraud Percent(Yes): 7.98%	7.99%	7.95%


```
set.seed(1235)
test_index <- sample(nrow(claims_df), 0.3*nrow(claims_df))
train_set <- claims_df[-test_index,]
test_set <- claims_df[test_index,]
```

Evaluation

LOGISTIC REGRESSION

Train Set

```
          Reference
Prediction   No  Yes
       No  4907    0
       Yes  303  471

               Accuracy : 0.9467
                 95% CI : (0.9405, 0.952
    No Information Rate : 0.9171
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7287

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 1.00000
            Specificity : 0.94184
         Pos Pred Value : 0.60853
         Neg Pred Value : 1.00000
             Prevalence : 0.08291
         Detection Rate : 0.08291
   Detection Prevalence : 0.13624
      Balanced Accuracy : 0.97092
```

Test Set

```
          Reference
Prediction   No  Yes
       No  2099    1
       Yes  126  194

               Accuracy : 0.9475
                 95% CI : (0.9379, 0.9561
    No Information Rate : 0.9194
    P-Value [Acc > NIR] : 4.539e-08

                  Kappa : 0.726

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.99487
            Specificity : 0.94337
         Pos Pred Value : 0.60625
         Neg Pred Value : 0.99952
             Prevalence : 0.08058
         Detection Rate : 0.08017
   Detection Prevalence : 0.13223
      Balanced Accuracy : 0.96912
```

Set seed value = 123

Train Set

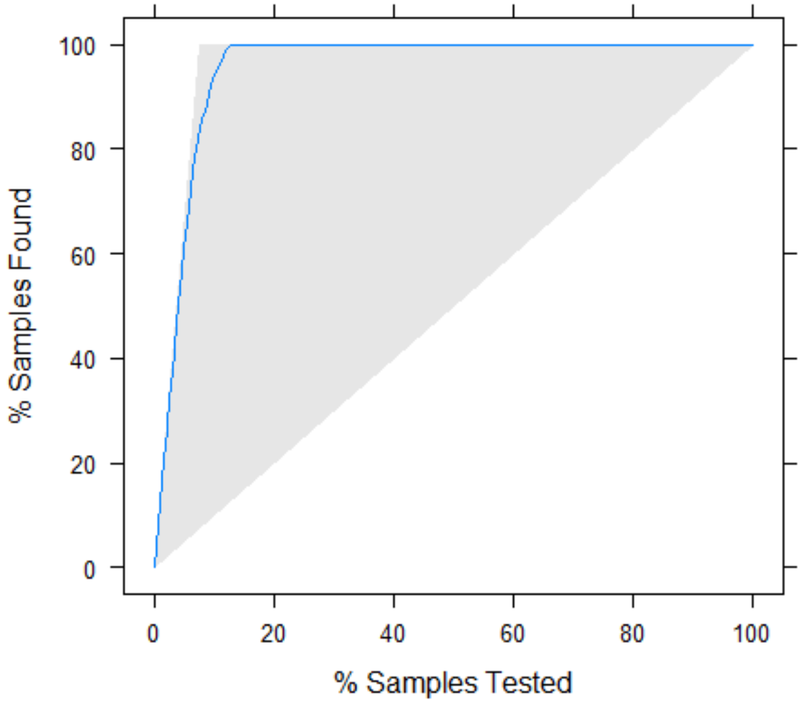
```
          Reference
Prediction   No  Yes
       No  4875    0
       Yes  330  467

               Accuracy : 0.9418
                 95% CI : (0.9354, 0.9478)
    No Information Rate : 0.9177
    P-Value [Acc > NIR] : 2.212e-12

                  Kappa : 0.7087

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 1.00000
            Specificity : 0.93660
         Pos Pred Value : 0.58595
         Neg Pred Value : 1.00000
             Prevalence : 0.08233
         Detection Rate : 0.08233
   Detection Prevalence : 0.14051
      Balanced Accuracy : 0.96830
```

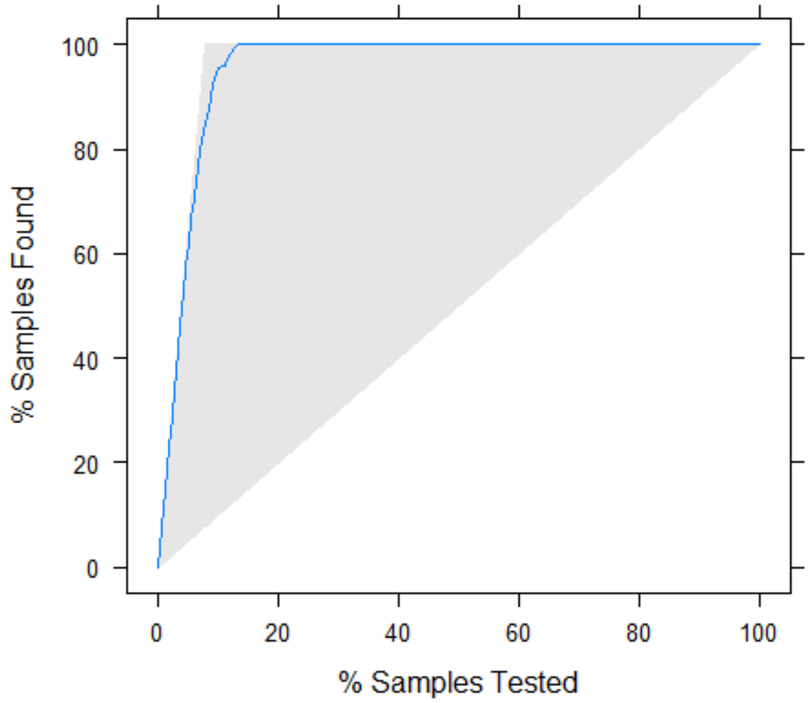
Test Set

```
          Reference
Prediction   No  Yes
       No  2133    5
       Yes   97  194

               Accuracy : 0.958
                 95% CI : (0.9493, 0.9656)
    No Information Rate : 0.9181
    P-Value [Acc > NIR] : 2.957e-15

                  Kappa : 0.7694

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.97487
            Specificity : 0.95650
         Pos Pred Value : 0.66667
         Neg Pred Value : 0.99766
             Prevalence : 0.08193
         Detection Rate : 0.07987
   Detection Prevalence : 0.11980
      Balanced Accuracy : 0.96569
```

Set seed value = 1235

LIFT





Set seed value = 123

Set seed value = 1235

TREE

Train Set

```
          Reference
Prediction   No  Yes
       No  5291   74
       Yes   77  397

               Accuracy : 0.9741
                 95% CI : (0.9697, 0.978
    No Information Rate : 0.9193
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8261

 Mcnemar's Test P-Value : 0.8707

            Sensitivity : 0.84289
            Specificity : 0.98566
         Pos Pred Value : 0.83755
         Neg Pred Value : 0.98621
             Prevalence : 0.08066
         Detection Rate : 0.06799
   Detection Prevalence : 0.08118
      Balanced Accuracy : 0.91427
```

Test Set

```
          Reference
Prediction   No  Yes
       No  2269   42
       Yes   38  153

               Accuracy : 0.968
                 95% CI : (0.9604, 0.9746)
    No Information Rate : 0.9221
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7754

 Mcnemar's Test P-Value : 0.7373

            Sensitivity : 0.78462
            Specificity : 0.98353
         Pos Pred Value : 0.80105
         Neg Pred Value : 0.98183
             Prevalence : 0.07794
         Detection Rate : 0.06115
   Detection Prevalence : 0.07634
      Balanced Accuracy : 0.88407
```

Set seed value = 123

Train Set

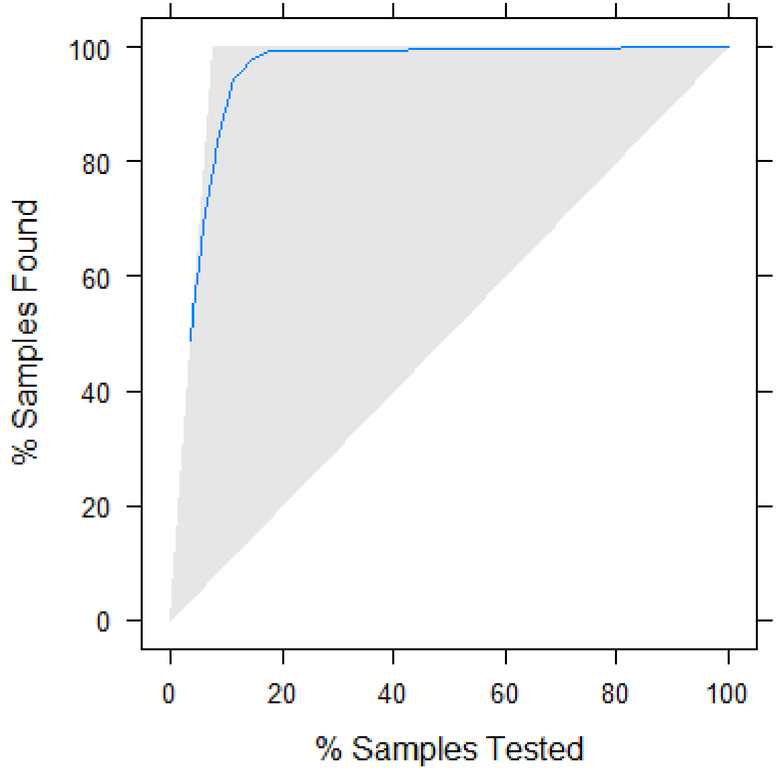
```
          Reference
Prediction   No  Yes
       No  5297   93
       Yes   75  374

               Accuracy : 0.9712
                 95% CI : (0.9666, 0.9754)
    No Information Rate : 0.92
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.801

 Mcnemar's Test P-Value : 0.1897

            Sensitivity : 0.80086
            Specificity : 0.98604
         Pos Pred Value : 0.83296
         Neg Pred Value : 0.98275
             Prevalence : 0.07998
         Detection Rate : 0.06405
   Detection Prevalence : 0.07690
      Balanced Accuracy : 0.89345
```

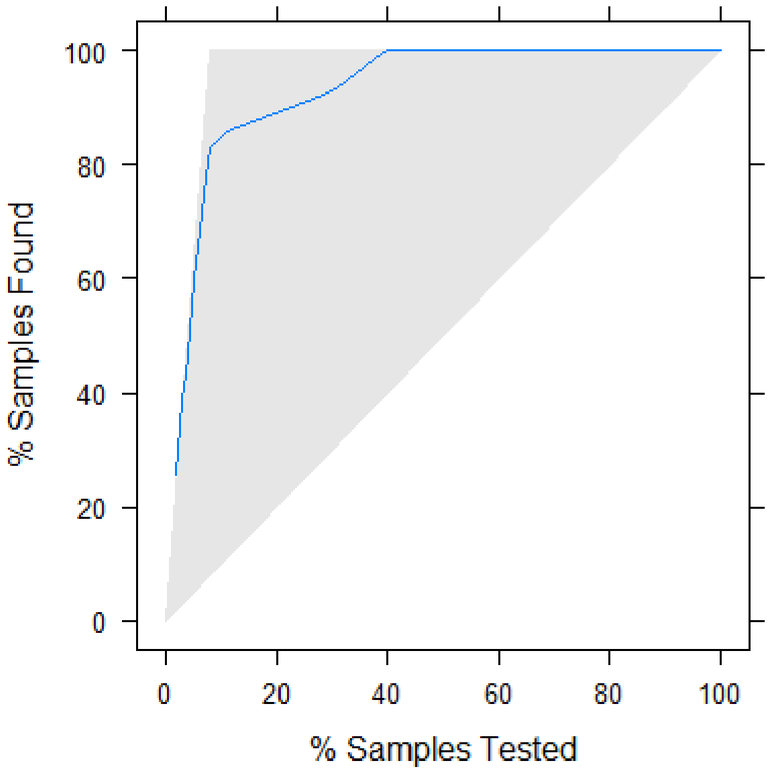
Test Set

```
          Reference
rediction   No  Yes
       No  2274   50
       Yes   29  149

               Accuracy : 0.9684
                 95% CI : (0.9608, 0.9749)
    No Information Rate : 0.9205
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.7734

 Mcnemar's Test P-Value : 0.02444

            Sensitivity : 0.74874
            Specificity : 0.98741
         Pos Pred Value : 0.83708
         Neg Pred Value : 0.97849
             Prevalence : 0.07954
         Detection Rate : 0.05955
   Detection Prevalence : 0.07114
      Balanced Accuracy : 0.86808
```

Set seed value = 1235

LIFT





Set seed value = 123

Set seed value = 1235

Conclusion

Detected Fraud

Detected Fraud

Know the relationship between each variable

Detected Fraud

Know the relationship between each variable

All the model

Result from Test Set using seed 1235

	Reference	
Prediction	No	Yes
No	2133	5
Yes	97	194

LOGISTIC REGRESSION

	Reference	
Prediction	No	Yes
No	2274	50
Yes	29	149

DECISION TREE

Non-Fraud

Misdetected Fraud

Undetected Fraud

Detected Fraud

LOGISTIC REGRESSION

	Reference	
Prediction	No	Yes
No	2133	5
Yes	97	194

DECISION TREE

	Reference	
Prediction	No	Yes
No	2274	50
Yes	29	149

Weighted with MONEY

LOGISTIC REGRESSION

Cost for re-check fraud = 97 = 97

Cost for product claim given that fraud = 5 = 20

Total = 117

DECISION TREE

Cost for re-check fraud = 29 = 29

Cost for product claim given that fraud = 50 = 200

Total = 229

LOGISTIC REGRESSION

	Reference	
Prediction	No	Yes
No	2133	5
Yes	97	194

DECISION TREE

	Reference	
Prediction	No	Yes
No	2274	50
Yes	29	149

Weighted with MONEY 

LOGISTIC REGRESSION

	Reference	
Prediction	No	Yes
No	2133	5
Yes	97	194

DECISION TREE

	Reference	
Prediction	No	Yes
No	2274	50
Yes	29	149

Weighted with MONEY

LOGISTIC REGRESSION

Cost for re-check fraud = 97 = 97

Cost for product claim given that fraud = 5 = 20

Total = 117

DECISION TREE

Cost for re-check fraud = 29 = 29

Cost for product claim given that fraud = 50 = 200

Total = 229

Logistic regression is the BEST model

Members

Chanatth	Sermsongsakulchai	62070503411
Chotiya	Pertpring	62070503413
Chatchapon	Sukitporn-Udom	62070503455
Pahsawee	Pansrithong	62070503459

THANK YOU