Marketing Strategies Review

2022-07-14

Reading Data

First 6 rows of the data table.

```
raw_sales_data = as.data.frame(read.csv("data_20160405.csv"))
head(raw_sales_data)
```

```
##   X      accID  accType accSize accTargets district    month   sales qty
## 1 1 XYZ-987002 Pharmacy     700         25        3 01-10-13       0   0
## 2 2 XYZ-987002 Pharmacy     700         25        3 01-11-13  852365   8
## 3 3 XYZ-987002 Pharmacy     700         25        3 01-12-13 2557096  25
## 4 4 XYZ-987002 Pharmacy     700         25        3 01-01-14 4261826  41
## 5 5 XYZ-987002 Pharmacy     700         25        3 01-02-14 5966556  57
## 6 6 XYZ-987002 Pharmacy     700         25        3 01-03-14 6818921  66
##   strategy1 strategy2 strategy3 salesVisit1 salesVisit2 salesVisit3 salesVisit4
## 1         0         0         0           0           0           0           0
## 2         0         0         0           0           0           0           0
## 3         0         0         0           0           0           0           0
## 4         0         0         0           0      825000      300000           0
## 5         0         0         0           0           0           0           0
## 6         0         0         0           0     1050000       75000           0
##   salesVisit5 compBrand
## 1           0         4
## 2       75000         4
## 3           0         4
## 4           0         4
## 5           0         4
## 6           0         4
```

Missing values

Number of missing values in each column

```
##           X       accID     accType     accSize  accTargets    district
##           0           0           0           0           0           0
##       month       sales         qty   strategy1   strategy2   strategy3
##           0           0           0           0           0           0
## salesVisit1 salesVisit2 salesVisit3 salesVisit4 salesVisit5   compBrand
##           0           0           0           0           0           0
```

Data preprocessing

There are 14 independent variables: 1. 3 categorical variables. 2. 10 numerical variables. 3. 1 date variable.

Categorical variables

Converting account type variable to dummies.

```
library(fastDummies)
library(knitr)

acc_type = data.frame(accType = raw_sales_data$accType)
acc_type_dummy = dummy_cols(acc_type)
acc_type_dummy = acc_type_dummy[, -1]
head(acc_type_dummy)
```

```
##   accType_Hospital accType_Pharmacy accType_Polyclinic accType_Private Clinic
## 1                0                1                  0                      0
## 2                0                1                  0                      0
## 3                0                1                  0                      0
## 4                0                1                  0                      0
## 5                0                1                  0                      0
## 6                0                1                  0                      0
```

Since there is only 1 district(3), we are going to ignore this variable.

```
print(paste("Unique district vales:", unique(raw_sales_data$district)))
```

```
## [1] "Unique district vales: 3"
```

Convert Competitive brands variable to binary.

```
comp_brand = data.frame(accType = raw_sales_data$compBrand)
comp_brand_dummy = dummy_cols(comp_brand)
comp_brand_dummy = comp_brand_dummy[, -1]
head(comp_brand_dummy)
```

```
##   accType_4 accType_5
## 1         1         0
## 2         1         0
## 3         1         0
## 4         1         0
## 5         1         0
## 6         1         0
```

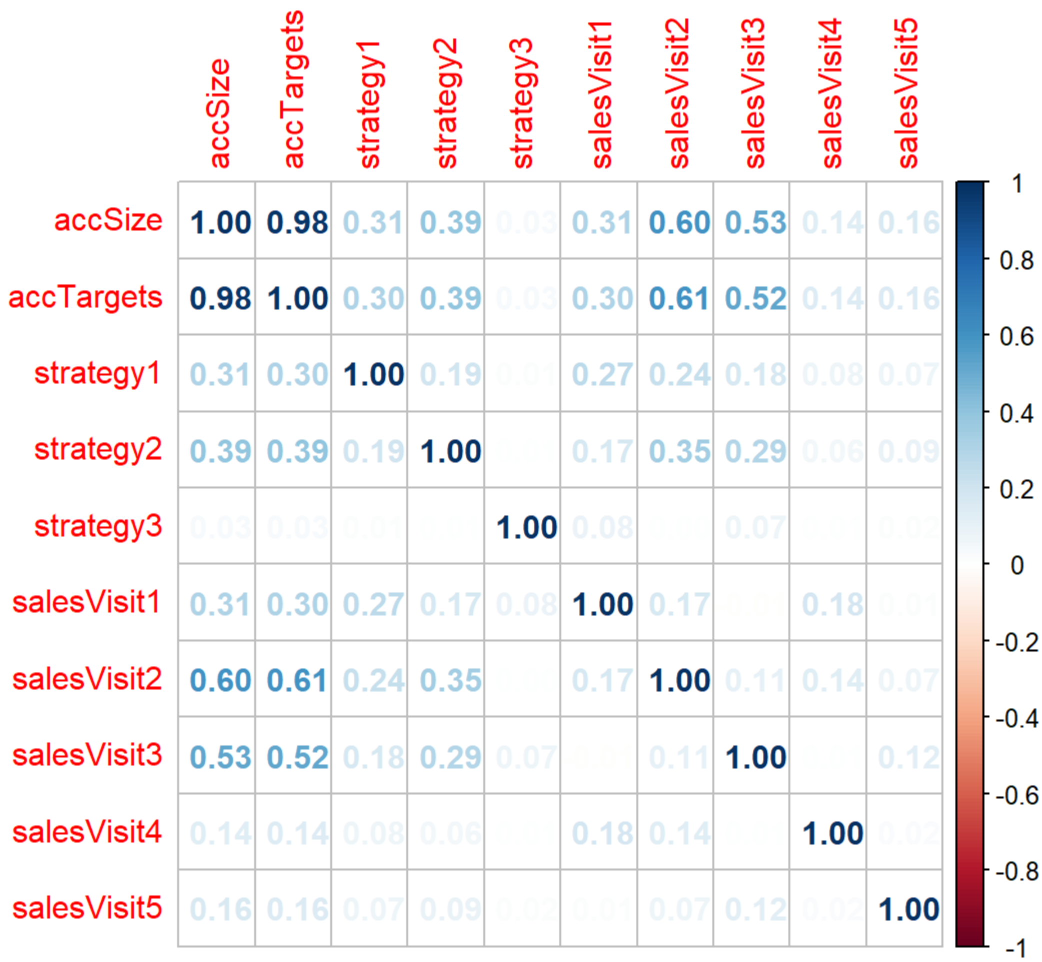
Numerical variables

As we can see from the correlation plot account size and account target are highly correlated. We should drop one of these variables.

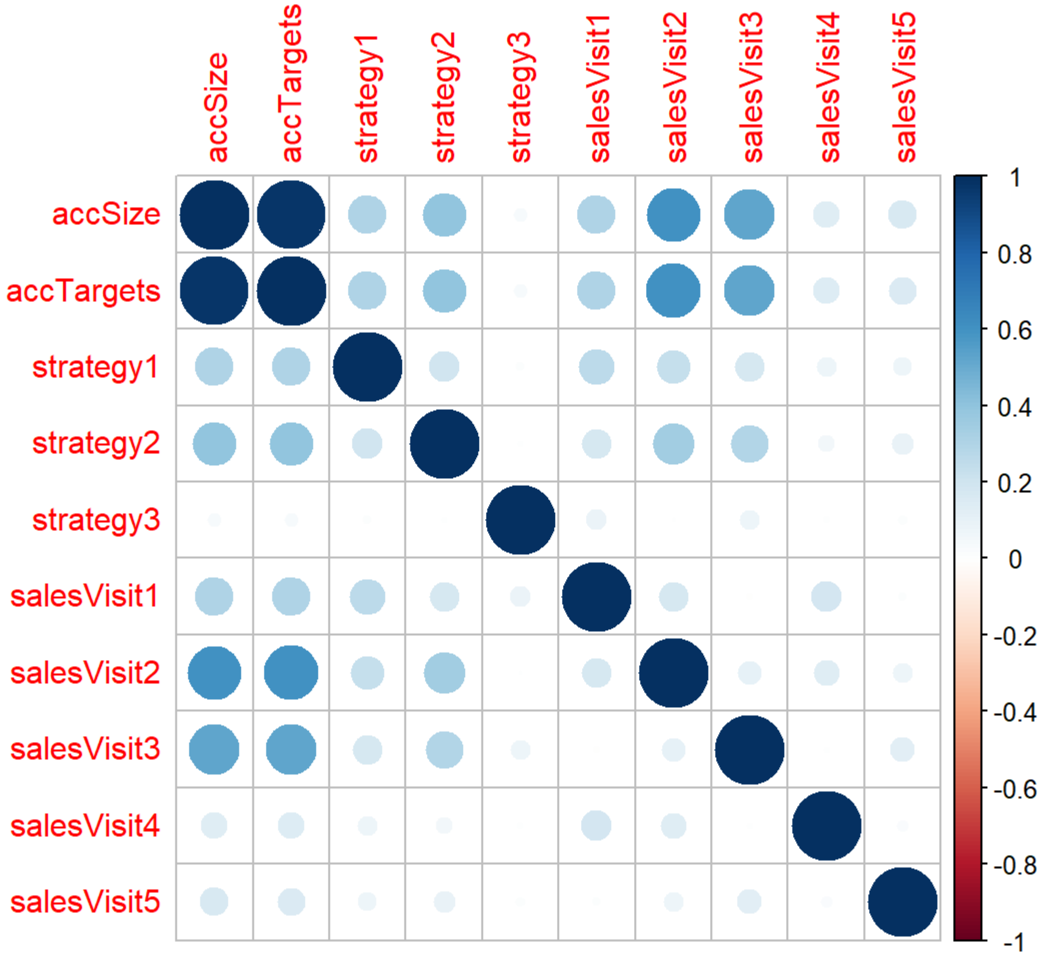
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
numerical_columns = colnames(raw_sales_data)[!colnames(raw_sales_data) %in% c("X", "accID", "acc
Type","district", "compBrand", "sales", "qty", "month")]
cormatrix = cor(raw_sales_data[numerical_columns] )
corrplot(cormatrix, method = "number")
```




```
corrplot(cormatrix, method = "circle")
```



###

Standardization of numeric variables We need to scale the numeric variables and remove account targets

```
numerical_columns_data = raw_sales_data[numerical_columns]
scaled_numerical_columns_data = scale(numerical_columns_data)

# Removing account size
scaled_numerical_columns_data = subset(scaled_numerical_columns_data, select = -c(accTargets) )
head(scaled_numerical_columns_data)
```

```
##        accSize  strategy1 strategy2   strategy3 salesVisit1 salesVisit2
## [1,] 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137  -0.5956210
## [2,] 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137  -0.5956210
## [3,] 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137  -0.5956210
## [4,] 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137   0.4486856
## [5,] 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137  -0.5956210
## [6,] 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137   0.7334965
##      salesVisit3 salesVisit4 salesVisit5
## [1,]  -0.5708616  -0.1828577  -0.1642669
## [2,]  -0.5708616  -0.1828577   3.4569055
## [3,]  -0.5708616  -0.1828577  -0.1642669
## [4,]  -0.1765288  -0.1828577  -0.1642669
## [5,]  -0.5708616  -0.1828577  -0.1642669
## [6,]  -0.4722784  -0.1828577  -0.1642669
```

Converting date variable into 2 variables of month and year

Month can have significant impact on sales if sales are seasonal. Year can be useful if there are any yearly changes in sales.

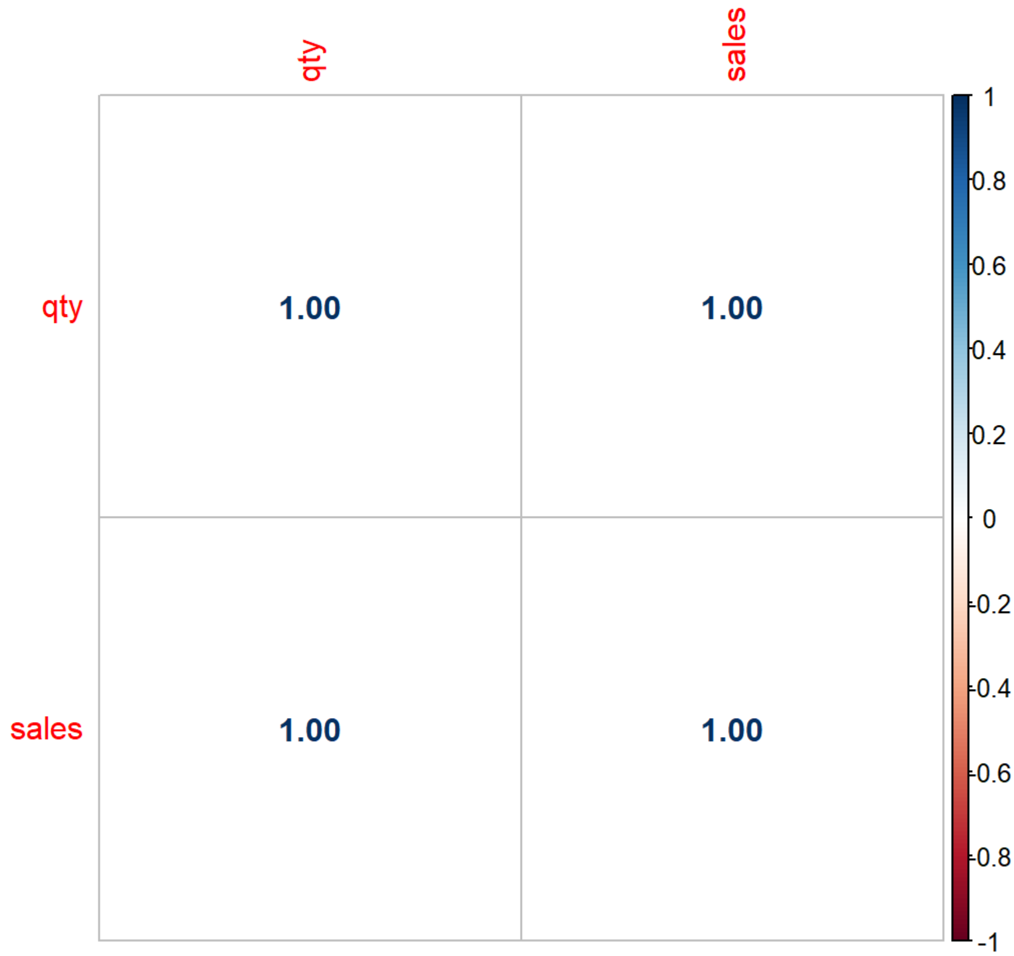
```
date_data        = as.POSIXct(raw_sales_data$month)
data_month       = data.frame(month = format(date_data,"%B") )
data_month_dummy = dummy_cols(data_month)
data_month_dummy = data_month_dummy[, -1]

data_year       = data.frame(year = as.numeric(format(date_data,"%Y") ) )
data_year       = data_year - min(data_year)

date_data_transformed =   cbind(data_month_dummy, data_year)
head(date_data_transformed, 10)
```

```
##    month_April month_August month_December month_February month_January
## 1            0            0              0              0             0
## 2            0            0              0              0             0
## 3            0            0              1              0             0
## 4            0            0              0              0             1
## 5            0            0              0              1             0
## 6            0            0              0              0             0
## 7            1            0              0              0             0
## 8            0            0              0              0             0
## 9            0            0              0              0             0
## 10           0            0              0              0             0
##    month_July month_June month_March month_May month_November month_October
## 1           0          0           0         0              0             1
## 2           0          0           0         0              1             0
## 3           0          0           0         0              0             0
## 4           0          0           0         0              0             0
## 5           0          0           0         0              0             0
## 6           0          0           1         0              0             0
## 7           0          0           0         0              0             0
## 8           0          0           0         1              0             0
## 9           0          1           0         0              0             0
## 10          1          0           0         0              0             0
##    month_September year
## 1                0    0
## 2                0    0
## 3                0    0
## 4                0    0
## 5                0    0
## 6                0    0
## 7                0    0
## 8                0    0
## 9                0    0
## 10               0    0
```

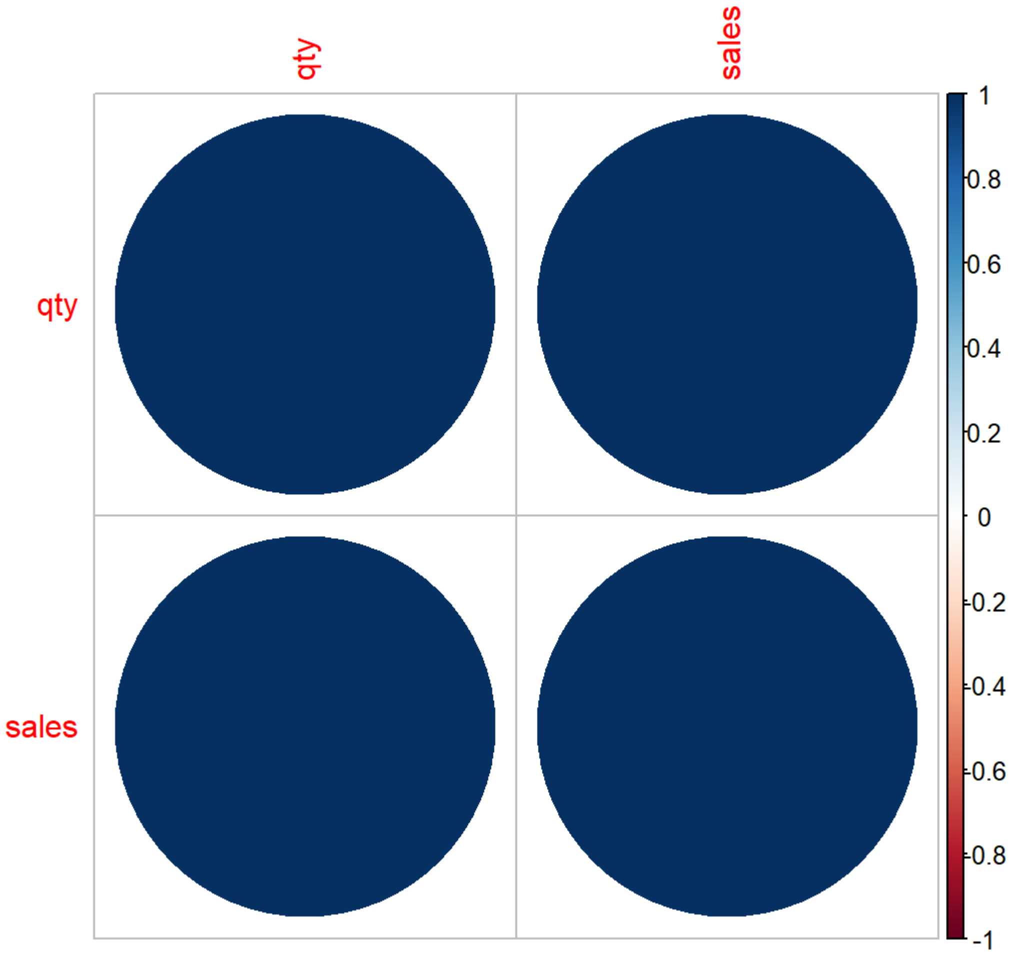
Checking correlation of dependent variables

Quantity and sales have correlation of 1. It means we need to consider only 1 of these.

```
dependent_data = raw_sales_data[, c("qty", "sales")]
cormatrix = cor(dependent_data)
corrplot(cormatrix, method = "number")
```



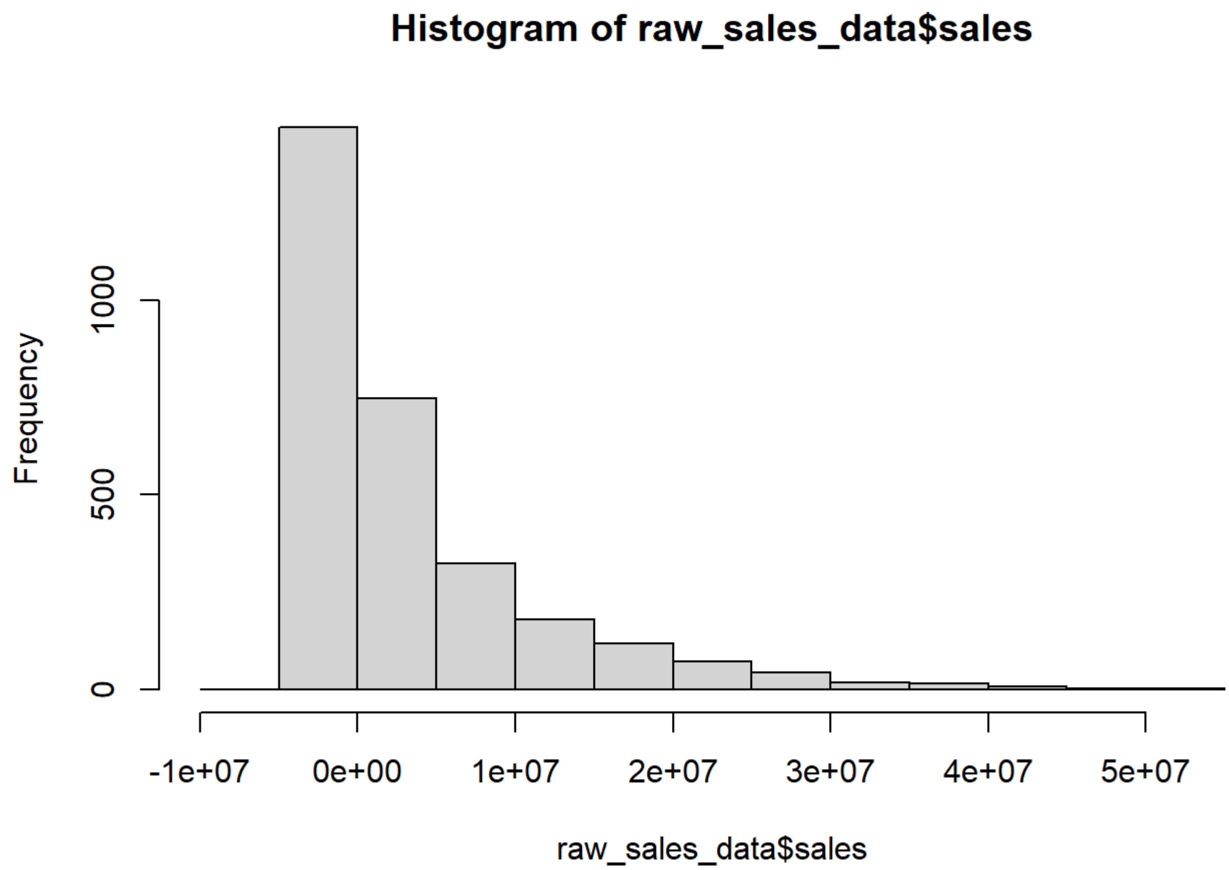

```
corrplot(cormatrix, method = "circle")
```



Distributioin of sales variable

There are some negative sales value. Maybe the customer returned some product that was either defective or was not used by the customer. We need to change these values to 0. Any strategy cannot have negative effect on the sales.

```
hist(raw_sales_data$sales)
```



New variable related to sales

We can create another dependent variable to mark whether sales have happened or not. This will be 0 if sales is 0 otherwise it will be 1.

```
sales_happened = data.frame(sales_happened = ifelse(raw_sales_data$sales > 0, 1, 0) )
head(sales_happened, 10)
```

```
##    sales_happened
## 1               0
## 2               1
## 3               1
## 4               1
## 5               1
## 6               1
## 7               1
## 8               1
## 9               1
## 10              1
```

Creating dataframe of predictors.

For categorical variables with more than 2 alternatives first column will be removed. If there are n alternatives for a categorical variable, n-1 columns are required to represent that variable.

```
acc_type_variable = acc_type_dummy[, -1]
comp_brand_variable = comp_brand_dummy[, -1]
date_data_variable  = date_data_transformed[colnames(date_data_transformed) != "month_April"]

logistic_data = cbind(acc_type_variable, comp_brand_variable, date_data_variable, scaled_numeric
al_columns_data, sales_happened)

head(logistic_data)
```

```
##   accType_Pharmacy accType_Polyclinic accType_Private Clinic
## 1                1                  0                      0
## 2                1                  0                      0
## 3                1                  0                      0
## 4                1                  0                      0
## 5                1                  0                      0
## 6                1                  0                      0
##   comp_brand_variable month_August month_December month_February month_January
## 1                   0            0              0              0             0
## 2                   0            0              0              0             0
## 3                   0            0              1              0             0
## 4                   0            0              0              0             1
## 5                   0            0              0              1             0
## 6                   0            0              0              0             0
##   month_July month_June month_March month_May month_November month_October
## 1          0          0           0         0              0             1
## 2          0          0           0         0              1             0
## 3          0          0           0         0              0             0
## 4          0          0           0         0              0             0
## 5          0          0           0         0              0             0
## 6          0          0           1         0              0             0
##   month_September year   accSize  strategy1 strategy2   strategy3 salesVisit1
## 1               0    0 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137
## 2               0    0 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137
## 3               0    0 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137
## 4               0    0 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137
## 5               0    0 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137
## 6               0    0 0.8043031 -0.1981688 -0.396749 -0.07770456  -0.1269137
##   salesVisit2 salesVisit3 salesVisit4 salesVisit5 sales_happened
## 1  -0.5956210  -0.5708616  -0.1828577  -0.1642669              0
## 2  -0.5956210  -0.5708616  -0.1828577   3.4569055              1
## 3  -0.5956210  -0.5708616  -0.1828577  -0.1642669              1
## 4   0.4486856  -0.1765288  -0.1828577  -0.1642669              1
## 5  -0.5956210  -0.5708616  -0.1828577  -0.1642669              1
## 6   0.7334965  -0.4722784  -0.1828577  -0.1642669              1
```

Logistic model to identify the influence of predictors on whether sale will happen or not.

```
library(nnet)

logistic_model = multinom(formula = sales_happened ~ ., data = logistic_data)
```

```
## # weights:  27 (26 variable)
## initial  value 2062.806009
## iter  10 value 1362.382523
## iter  20 value 1327.099882
## iter  30 value 1323.519978
## final  value 1323.518849
## converged
```

```
results_log = data.frame(coefficients = summary(logistic_model)$coefficients)
```

```
## Warning in sqrt(diag(vc)): NaNs produced
```

```
results_log$stdev        = summary(logistic_model)$standard.errors
```

```
## Warning in sqrt(diag(vc)): NaNs produced
```

```
results_log$zvalues      = results_log$coefficients / results_log$stdev
results_log$pvalues      = 2 * pnorm(-abs(results_log$zvalues) )

results_log = results_log[order(-abs(results_log$coefficients)),  ]
results_log
```

```
##                            coefficients      stdev     zvalues      pvalues
## comp_brand_variable         1.56753571 0.17421921  8.99749080 2.309352e-19
## accType_Polyclinic         -1.15488114 0.12585992 -9.17592444 4.477162e-20
## salesVisit2                 0.84429209 0.10171406  8.30064277 1.035496e-16
## `accType_Private Clinic`   -0.81981224 0.17669492 -4.63970470 3.489074e-06
## accSize                     0.80400366 0.11269605  7.13426634 9.730476e-13
## (Intercept)                 0.77842589 0.18008577  4.32252854 1.542511e-05
## salesVisit3                 0.66124001 0.09104173  7.26304339 3.784765e-13
## salesVisit1                 0.64687362 0.22075009  2.93034359 3.385874e-03
## month_October              -0.39527818 0.24704048 -1.60005426 1.095865e-01
## month_June                 -0.38167766 0.25462622 -1.49897232 1.338808e-01
## strategy2                   0.35754590 0.08428396  4.24215808 2.213807e-05
## month_March                -0.34005730 0.24427153 -1.39212822 1.638836e-01
## accType_Pharmacy           -0.31276857 0.18215033 -1.71709039 8.596266e-02
## month_May                   0.28531002 0.23889188  1.19430606 2.323583e-01
## month_September             0.24115830 0.25414198  0.94891170 3.426655e-01
## month_February             -0.23780861 0.24018507 -0.99010571 3.221225e-01
## month_November             -0.23209088 0.24351734 -0.95307744 3.405508e-01
## month_July                 -0.22040066 0.25813546 -0.85381783 3.932059e-01
## month_January              -0.19429592 0.24020587 -0.80887251 4.185885e-01
## strategy3                   0.17236882 0.14401308  1.19689701 2.313467e-01
## salesVisit5                 0.07273077 0.05888656  1.23509965 2.167934e-01
## salesVisit4                 0.05706541 0.06275946  0.90927183 3.632067e-01
## strategy1                   0.04580250 0.07815365  0.58605708 5.578372e-01
## month_December              0.03721688 0.23963811  0.15530453 8.765812e-01
## month_August                0.01084453 0.25594255  0.04237096 9.662030e-01
## year                        0.00000000        NaN         NaN          NaN
```

Confusian matrix of logistic model

```
# Predicting on test data'
prediction = predict(logistic_model, newdata = logistic_data)

# Confusion Matrix
cm = table(logistic_data$sales_happened, prediction)
cm
```

```
##    prediction
##        0    1
##   0 1179  264
##   1  350 1183
```

Strategy results

Effect of strategy on probability of sales.

```
strategy_results = results_log[c("strategy1", "strategy2", "strategy3"),  ]
strategy_results
```

```
##           coefficients      stdev   zvalues      pvalues
## strategy1    0.0458025 0.07815365 0.5860571 5.578372e-01
## strategy2    0.3575459 0.08428396 4.2421581 2.213807e-05
## strategy3    0.1723688 0.14401308 1.1968970 2.313467e-01
```

Effect of new entrant on probability sales

Effect of new entrant on whether sale will happen or not has very high p-value. It means according to the model new entrant does not have significant effect on whether sale will happen or not.

```
new_entrant_results = results_log[c("comp_brand_variable"),  ]
new_entrant_results
```

```
##                     coefficients     stdev  zvalues      pvalues
## comp_brand_variable     1.567536 0.1742192 8.997491 2.309352e-19
```

Decrease in probability of sales after entry of new competitor.

```
new_entrant_data = logistic_data[logistic_data$comp_brand_variable == 1, ]

new_entrant_prediction = predict(logistic_model, newdata = new_entrant_data, type = "prob")

without_new_entrant_data = new_entrant_data
without_new_entrant_data$comp_brand_variable = 0

without_new_entrant_prediction = predict(logistic_model, newdata = without_new_entrant_data, typ
e = "prob")

change_in_prob = (mean(new_entrant_prediction) - mean(without_new_entrant_prediction)) / mean(wi
thout_new_entrant_prediction) * 100

change_in_prob
```

```
## [1] 44.59279
```

```
data.frame(Condition = c("Without new competitor", "With new competitor"),
           `Average Probability` = c(mean(without_new_entrant_prediction), mean(new_entrant_pred
iction) )  )
```

```
##                Condition Average.Probability
## 1 Without new competitor           0.5033603
## 2    With new competitor           0.7278227
```

Creating data for linear regression on cases where sale has happened

```
log_sales = data.frame(sales = raw_sales_data$sales[logistic_model$fitted.values > 0.5] )


linear_data = cbind(logistic_data[logistic_model$fitted.values > 0.5, ], log_sales)
linear_data$sales[linear_data$sales < 0] = 0
linear_data = linear_data[, colnames(linear_data) != "sales_happened"]
head(linear_data)
```

```
##   accType_Pharmacy accType_Polyclinic accType_Private Clinic
## 3                1                  0                      0
## 4                1                  0                      0
## 6                1                  0                      0
## 7                1                  0                      0
## 8                1                  0                      0
## 9                1                  0                      0
##   comp_brand_variable month_August month_December month_February month_January
## 3                   0            0              1              0             0
## 4                   0            0              0              0             1
## 6                   0            0              0              0             0
## 7                   0            0              0              0             0
## 8                   0            0              0              0             0
## 9                   0            0              0              0             0
##   month_July month_June month_March month_May month_November month_October
## 3          0          0           0         0              0             0
## 4          0          0           0         0              0             0
## 6          0          0           1         0              0             0
## 7          0          0           0         0              0             0
## 8          0          0           0         1              0             0
## 9          0          1           0         0              0             0
##   month_September year   accSize  strategy1   strategy2   strategy3 salesVisit1
## 3               0    0 0.8043031 -0.1981688 -0.39674902 -0.07770456  -0.1269137
## 4               0    0 0.8043031 -0.1981688 -0.39674902 -0.07770456  -0.1269137
## 6               0    0 0.8043031 -0.1981688 -0.39674902 -0.07770456  -0.1269137
## 7               0    0 0.8043031 -0.1981688 -0.39674902 -0.07770456  -0.1269137
## 8               0    0 0.8043031 -0.1981688  0.00624825 -0.07770456  -0.1269137
## 9               0    0 0.8043031  0.3326429  0.35019601 -0.07770456  -0.1269137
##   salesVisit2 salesVisit3 salesVisit4 salesVisit5   sales
## 3  -0.5956210  -0.5708616  -0.1828577  -0.1642669 2557096
## 4   0.4486856  -0.1765288  -0.1828577  -0.1642669 4261826
## 6   0.7334965  -0.4722784  -0.1828577  -0.1642669 6818921
## 7   0.7334965   0.2178039  -0.1828577  -0.1642669 4261826
## 8   0.4486856   0.5135535  -0.1828577  -0.1642669 4261826
## 9   0.1638747   0.7107199  -0.1828577  -0.1642669 4261826
```

Linear regression model

```
linear_model = lm(formula = sales ~ ., data = linear_data)

results = data.frame(coefficients = summary(linear_model)$coefficients)
results = results[order(-abs(results$coefficients.Estimate)),  ]
results
```

```
##                             coefficients.Estimate coefficients.Std..Error
## comp_brand_variable                   6127424.699                605159.8
## accType_Polyclinic                   -5228143.512               1054740.5
## (Intercept)                           4625247.407                641091.5
## `accType_Private Clinic`             -2772844.312               1265493.5
## month_August                         -2534567.093                907813.7
## month_June                           -2454156.867                947166.8
## salesVisit2                           2301362.831                197186.1
## month_July                           -2236559.764                908900.6
## accSize                               2183982.914                305281.1
## accType_Pharmacy                     -2024684.759                572118.7
## month_March                          -2009403.182                885384.3
## month_October                        -1797405.455               1053371.7
## salesVisit3                           1351349.079                189760.0
## month_September                      -1319307.491                892955.8
## month_February                       -1265303.420                911336.1
## strategy2                             1240831.007                151917.9
## month_December                        1226393.169                967791.4
## salesVisit4                            999087.787                134596.3
## salesVisit1                            659324.437                148232.1
## month_November                        -233640.935               1017853.9
## month_May                             -166917.488                865229.2
## strategy1                              158818.676                143833.2
## month_January                           54362.077                942965.8
## strategy3                               20215.671                133509.5
## salesVisit5                             -2773.849                137807.3
##                             coefficients.t.value coefficients.Pr...t..
## comp_brand_variable                  10.12530042          2.576648e-23
## accType_Polyclinic                   -4.95680559          8.026343e-07
## (Intercept)                           7.21464513          8.763665e-13
## `accType_Private Clinic`             -2.19111695          2.860512e-02
## month_August                         -2.79194634          5.309324e-03
## month_June                           -2.59105041          9.666251e-03
## salesVisit2                          11.67101686          4.031856e-30
## month_July                           -2.46073078          1.398357e-02
## accSize                               7.15400523          1.344837e-12
## accType_Pharmacy                     -3.53892411          4.146987e-04
## month_March                          -2.26952659          2.338552e-02
## month_October                        -1.70633540          8.816400e-02
## salesVisit3                           7.12135811          1.691286e-12
## month_September                      -1.47746111          1.397735e-01
## month_February                       -1.38840474          1.652313e-01
## strategy2                             8.16777119          6.863630e-16
## month_December                        1.26720817          2.052884e-01
## salesVisit4                           7.42284898          1.965417e-13
## salesVisit1                           4.44791815          9.345635e-06
## month_November                       -0.22954270          8.184801e-01
## month_May                            -0.19291708          8.470515e-01
## strategy1                             1.10418683          2.696989e-01
## month_January                         0.05765010          9.540354e-01
## strategy3                             0.15141743          8.796679e-01
## salesVisit5                          -0.02012847          9.839437e-01
```

Linear model tests

Linear model passes all the normality tests with pvalues less than 0.05.

```
library(olsrr)
```

```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
ols_test_normality(linear_model)
```

```
## Warning in ks.test.default(y, "pnorm", mean(y), sd(y)): ties should not be
## present for the Kolmogorov-Smirnov test
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9348         0.0000
## Kolmogorov-Smirnov        0.1018         0.0000
## Cramer-von Mises         134.2255        0.0000
## Anderson-Darling         28.5071         0.0000
## -----------------------------------------------
```

Strategy results

```
strategy_results = results[c("strategy1", "strategy2", "strategy3"), ]
strategy_results
```

```
##           coefficients.Estimate coefficients.Std..Error coefficients.t.value
## strategy1             158818.68                143833.2            1.1041868
## strategy2            1240831.01                151917.9            8.1677712
## strategy3              20215.67                133509.5            0.1514174
##           coefficients.Pr...t..
## strategy1          2.696989e-01
## strategy2          6.863630e-16
## strategy3          8.796679e-01
```

Effect of new entrant on sales

According to model new entrant has negative effect on sales.

```
new_entrant_results = results[c("comp_brand_variable"),  ]
new_entrant_results
```

```
##                       coefficients.Estimate coefficients.Std..Error
## comp_brand_variable               6127425                605159.8
##                       coefficients.t.value coefficients.Pr...t..
## comp_brand_variable              10.1253          2.576648e-23
```

Decrease in sale after entry of new competitor.

```
new_entrant_data_linear = linear_data[linear_data$comp_brand_variable == 1, ]

new_entrant_prediction_sales = predict(linear_model, newdata = new_entrant_data_linear, type =
"response")
```

```
## Warning in predict.lm(linear_model, newdata = new_entrant_data_linear, type =
## "response"): prediction from a rank-deficient fit may be misleading
```

```
without_new_entrant_data_linear = new_entrant_data_linear
without_new_entrant_data_linear$comp_brand_variable = 0

without_new_entrant_prediction_sales = predict(linear_model, newdata = without_new_entrant_data_
linear, type = "response")
```

```
## Warning in predict.lm(linear_model, newdata = without_new_entrant_data_linear, :
## prediction from a rank-deficient fit may be misleading
```

```
change_in_sales = (mean(new_entrant_prediction_sales) - mean(without_new_entrant_prediction_sale
s)) / mean(without_new_entrant_prediction_sales) * 100

change_in_sales
```

```
## [1] 161.4796
```

```
data.frame(Condition = c("Without new competitor", "With new competitor"),
           `Average Sales` = c(mean(without_new_entrant_prediction_sales), mean(new_entrant_pred
iction_sales) )  )
```

```
##                Condition Average.Sales
## 1 Without new competitor       3794551
## 2    With new competitor       9921976
```

Increase in sales by increasing expenditure on strategy 1 by 1%

```
scaled_increased_variable = function(variable) {
    mean_variable = mean(variable)
    sd_variable   = sd(variable)

    increased_variable        = variable * 1.01
    scaled_increased_variable = (increased_variable - mean_variable) / sd_variable

    return(scaled_increased_variable)
}

mean_sales_predicted = mean(linear_model$fitted.values)

# Increase in sales by increasing expenditure in strategy 1 by 1%
scaled_increased_s1 = scaled_increased_variable(numerical_columns_data$strategy1)

linear_data_s1_increased = linear_data
linear_data_s1_increased$strategy1 = scaled_increased_s1[logistic_model$fitted.values > 0.5]

sales_predicted_increased_s1 = predict(object = linear_model, newdata = linear_data_s1_increase
d)
```

```
## Warning in predict.lm(object = linear_model, newdata =
## linear_data_s1_increased): prediction from a rank-deficient fit may be
## misleading
```

```
mean_sales_predicted_increased_s1 =  mean(sales_predicted_increased_s1)

increased_sales_increased_s1 = (mean_sales_predicted_increased_s1 - mean_sales_predicted) / mean
_sales_predicted

# Increase in sales by increasing expenditure in strategy 2 by 1%
scaled_increased_s2 = scaled_increased_variable(numerical_columns_data$strategy2)

linear_data_s2_increased = linear_data
linear_data_s2_increased$strategy2 = scaled_increased_s2[logistic_model$fitted.values > 0.5]

sales_predicted_increased_s2 = predict(object = linear_model, newdata = linear_data_s2_increase
d)
```

```
## Warning in predict.lm(object = linear_model, newdata =
## linear_data_s2_increased): prediction from a rank-deficient fit may be
## misleading
```

```
mean_sales_predicted_increased_s2 =  mean(sales_predicted_increased_s2)

increased_sales_increased_s2 = (mean_sales_predicted_increased_s2 - mean_sales_predicted) / mean
_sales_predicted

# Increase in sales by increasing expenditure in strategy 3 by 1%
scaled_increased_s3 = scaled_increased_variable(numerical_columns_data$strategy3)

linear_data_s3_increased = linear_data
linear_data_s3_increased$strategy3 = scaled_increased_s3[logistic_model$fitted.values > 0.5]

sales_predicted_increased_s3 = predict(object = linear_model, newdata = linear_data_s3_increase
d)
```

```
## Warning in predict.lm(object = linear_model, newdata =
## linear_data_s3_increased): prediction from a rank-deficient fit may be
## misleading
```

```
mean_sales_predicted_increased_s3 =  mean(sales_predicted_increased_s3)

increased_sales_increased_s3 = (mean_sales_predicted_increased_s3 - mean_sales_predicted) / mean
_sales_predicted




sale_increase_by_strategy = data.frame(strategy = c("Strategy 1", "Strategy 2", "Strategy 3"),
                                  `Increase in sales (%)` = c(increased_sales_increased_s1, incr
eased_sales_increased_s2,
                                                          increased_sales_increased_s3) *100, ch
eck.names = FALSE  )
sale_increase_by_strategy
```

```
##     strategy Increase in sales (%)
## 1 Strategy 1          0.0076091887
## 2 Strategy 2          0.1202930320
## 3 Strategy 3          0.0004027859
```

Increase in probability of sales by increasing expenditure of each strategy by 1% (one at a time)

```
mean_prob_predicted       = mean(logistic_model$fitted.values)

# Increase in probability of sales by increasing expenditure in strategy 1 by 1%
logistic_data_s1_increased = logistic_data
logistic_data_s1_increased$strategy1 = scaled_increased_s1

prob_predicted_increased_s1 = predict(object = logistic_model, newdata = logistic_data_s1_increa
sed, type = "prob")
mean_prob_predicted_increased_s1 =  mean(prob_predicted_increased_s1)

increased_prob_increased_s1 = (mean_prob_predicted_increased_s1 - mean_prob_predicted) / mean_pr
ob_predicted

# Increase in probability of sales by increasing expenditure in strategy 2 by 1%
logistic_data_s2_increased = logistic_data
logistic_data_s2_increased$strategy2 = scaled_increased_s2

prob_predicted_increased_s2 = predict(object = logistic_model, newdata = logistic_data_s2_increa
sed, type = "prob")
mean_prob_predicted_increased_s2 =  mean(prob_predicted_increased_s2)

increased_prob_increased_s2 = (mean_prob_predicted_increased_s2 - mean_prob_predicted) / mean_pr
ob_predicted

# Increase in probability of sales by increasing expenditure in strategy 2 by 1%
logistic_data_s3_increased = logistic_data
logistic_data_s3_increased$strategy3 = scaled_increased_s3

prob_predicted_increased_s3 = predict(object = logistic_model, newdata = logistic_data_s3_increa
sed, type = "prob")
mean_prob_predicted_increased_s3 =  mean(prob_predicted_increased_s3)

increased_prob_increased_s3 = (mean_prob_predicted_increased_s3 - mean_prob_predicted) / mean_pr
ob_predicted

prob_increase_by_strategy = data.frame(strategy = c("Strategy 1", "Strategy 2", "Strategy 3"),
                                  `Increase in probability of sale (%)` = c(increased_prob_incre
ased_s1,
                                                                            increased_prob_incr
eased_s2,
                                                                            increased_prob_incr
eased_s3) *100,
                                  check.names = FALSE  )
prob_increase_by_strategy
```

```
##     strategy Increase in probability of sale (%)
## 1 Strategy 1                        0.0013614076
## 2 Strategy 2                        0.0187656498
## 3 Strategy 3                        0.0008654806
```

Regression of each account ID

```
account_ids = unique(raw_sales_data$accID)

new_d = cbind(logistic_data, sales = raw_sales_data$sales)
new_d = new_d[, colnames(new_d) != "sales_happened"]
new_d$sales[new_d$sales < 0] = 0

coeff_matrix = matrix(NA, nrow = length(account_ids), ncol = 3)
rownames(coeff_matrix) = account_ids
colnames(coeff_matrix) = c("1 coefficient", "2 coefficient", "3 coefficient")

pvalue_matrix = matrix(NA, nrow = length(account_ids), ncol = 3)
rownames(pvalue_matrix) = account_ids
colnames(pvalue_matrix) = c("1 p-value", "2 p-value", "3 p-value")


for(a in 1:length(account_ids) ) {
  ac_data = new_d[raw_sales_data$accID == as.character(account_ids[a]), ]
  drop = NULL
  for (c in 1:ncol(ac_data) ) {
    if (length(unique(ac_data[, c] ) ) == 1 )
      drop = c(drop, c)
  }
  ac_data = ac_data[, -drop]

  ac_model = lm(formula = sales ~ ., data = ac_data)

  m_summary = summary(ac_model)$coefficients

  if ("strategy1" %in% rownames(m_summary) ) {
    coeff_matrix[a, 1 ] = c(m_summary["strategy1", 1] )
    pvalue_matrix[a, 1 ] = c(m_summary["strategy1", 4] )
  }
  if ("strategy2" %in% rownames(m_summary) ) {
    coeff_matrix[a, 2 ] = c(m_summary["strategy2", 1] )
    pvalue_matrix[a, 2 ] = c(m_summary["strategy2", 4] )
  }
  if ("strategy3" %in% rownames(m_summary) ) {
    coeff_matrix[a, 3 ] = c(m_summary["strategy3", 1] )
    pvalue_matrix[a, 3 ] = c(m_summary["strategy3", 4] )
  }

}
```

```
## Warning in summary.lm(ac_model): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(ac_model): essentially perfect fit: summary may be
## unreliable
```

```
head(coeff_matrix)
```

```
##             1 coefficient 2 coefficient 3 coefficient
## XYZ-987002       3967918     1503745.3            NA
## XYZ-987005       4910557      971457.5            NA
## XYZ-987006      -4969514     -387764.7            NA
## XYZ-987007       7270050     7626424.5            NA
## XYZ-987008      -1695666     3489203.0            NA
## XYZ-987009       3361677     1185798.7            NA
```

```
head(pvalue_matrix)
```

```
##              1 p-value  2 p-value 3 p-value
## XYZ-987002 0.67550916 0.14810405        NA
## XYZ-987005 0.74068451 0.88952908        NA
## XYZ-987006 0.04072497 0.62234879        NA
## XYZ-987007 0.02103019 0.04041024        NA
## XYZ-987008 0.67359894 0.54560341        NA
## XYZ-987009 0.23651807 0.21549262        NA
```

Best strategy for each client

```
best_strategy = coeff_matrix
#best_strategy[is.na(best_strategy)] = -Inf

bs = as.matrix(apply(best_strategy, 1, max, na.rm = TRUE) )
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf
```

```r
bs_result = NULL

for (r in 1:nrow(best_strategy) ) {
  bs_value  = max(best_strategy[r, ], na.rm = TRUE )
  bs_result = c(bs_result, match(bs_value, best_strategy[r, ] ) )

}
```

```
## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf
```

```
## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf

## Warning in max(best_strategy[r, ], na.rm = TRUE): no non-missing arguments to
## max; returning -Inf
```

```
bs_result_data_frame = data.frame(accID = rownames(best_strategy), `Best strategy` = paste("Stra
tegy", bs_result), check.names = FALSE )

bs_result_data_frame
```

```
##          accID Best strategy
## 1   XYZ-987002    Strategy 1
## 2   XYZ-987005    Strategy 1
## 3   XYZ-987006    Strategy 2
## 4   XYZ-987007    Strategy 2
## 5   XYZ-987008    Strategy 2
## 6   XYZ-987009    Strategy 1
## 7   XYZ-987010    Strategy 2
## 8   XYZ-987014   Strategy NA
## 9   XYZ-987015    Strategy 1
## 10  XYZ-987016    Strategy 1
## 11  XYZ-987017    Strategy 2
## 12  XYZ-987019    Strategy 2
## 13  XYZ-987021    Strategy 3
## 14  XYZ-987022    Strategy 2
## 15  XYZ-987023    Strategy 2
## 16  XYZ-987024    Strategy 2
## 17  XYZ-987025    Strategy 1
## 18  XYZ-987026    Strategy 1
## 19  XYZ-987027    Strategy 2
## 20  XYZ-987029    Strategy 2
## 21  XYZ-987030    Strategy 1
## 22  XYZ-987032    Strategy 1
## 23  XYZ-987033    Strategy 1
## 24  XYZ-987034    Strategy 1
## 25  XYZ-987035    Strategy 3
## 26  XYZ-987036    Strategy 2
## 27  XYZ-987037    Strategy 3
## 28  XYZ-987039    Strategy 2
## 29  XYZ-987043    Strategy 2
## 30  XYZ-987045    Strategy 1
## 31  XYZ-987047   Strategy NA
## 32  XYZ-987049    Strategy 1
## 33  XYZ-987050    Strategy 2
## 34  XYZ-987052    Strategy 3
## 35  XYZ-987054    Strategy 1
## 36  XYZ-987056    Strategy 2
## 37  XYZ-987059    Strategy 2
## 38  XYZ-987060   Strategy NA
## 39  XYZ-987061    Strategy 1
## 40  XYZ-987062    Strategy 2
## 41  XYZ-987063    Strategy 2
## 42  XYZ-987065    Strategy 2
## 43  XYZ-987067    Strategy 2
## 44  XYZ-987069    Strategy 2
## 45  XYZ-987070    Strategy 2
## 46  XYZ-987071    Strategy 3
## 47  XYZ-987072    Strategy 2
## 48  XYZ-987075    Strategy 2
## 49  XYZ-987077    Strategy 2
## 50  XYZ-987079    Strategy 1
## 51  XYZ-987081    Strategy 2
```

```
## 52  XYZ-987082    Strategy 1
## 53  XYZ-987083    Strategy 2
## 54  XYZ-987085    Strategy 2
## 55  XYZ-987086    Strategy 2
## 56  XYZ-987087    Strategy 2
## 57  XYZ-987088    Strategy 2
## 58  XYZ-987089    Strategy 2
## 59  XYZ-987090    Strategy 2
## 60  XYZ-987091   Strategy NA
## 61  XYZ-987092    Strategy 2
## 62  XYZ-987093   Strategy NA
## 63  XYZ-987095   Strategy NA
## 64  XYZ-987096    Strategy 1
## 65  XYZ-987097    Strategy 2
## 66  XYZ-987100    Strategy 1
## 67  XYZ-987102    Strategy 1
## 68  XYZ-987103    Strategy 2
## 69  XYZ-987104    Strategy 2
## 70  XYZ-987106   Strategy NA
## 71  XYZ-987107    Strategy 2
## 72  XYZ-987109    Strategy 2
## 73  XYZ-987111    Strategy 2
## 74  XYZ-987112   Strategy NA
## 75  XYZ-987113   Strategy NA
## 76  XYZ-987118    Strategy 2
## 77  XYZ-987119    Strategy 2
## 78  XYZ-987120    Strategy 2
## 79  XYZ-987121   Strategy NA
## 80  XYZ-987124    Strategy 3
## 81  XYZ-987127    Strategy 2
## 82  XYZ-987130    Strategy 2
## 83  XYZ-987133    Strategy 2
## 84  XYZ-987134    Strategy 2
## 85  XYZ-987135    Strategy 3
## 86  XYZ-987136    Strategy 2
## 87  XYZ-987137    Strategy 1
## 88  XYZ-987138    Strategy 1
## 89  XYZ-987139    Strategy 2
## 90  XYZ-987141    Strategy 1
## 91  XYZ-987143    Strategy 1
## 92  XYZ-987144    Strategy 2
## 93  XYZ-987145    Strategy 2
## 94  XYZ-987146    Strategy 1
## 95  XYZ-987147    Strategy 2
## 96  XYZ-987148    Strategy 2
## 97  XYZ-987149    Strategy 2
## 98  XYZ-987150    Strategy 2
## 99  XYZ-987151    Strategy 2
## 100 XYZ-987153   Strategy NA
## 101 XYZ-987156    Strategy 3
## 102 XYZ-987157   Strategy NA
## 103 XYZ-987160    Strategy 1
```

```
## 104 XYZ-987161    Strategy 1
## 105 XYZ-987163    Strategy 3
## 106 XYZ-987164    Strategy 2
## 107 XYZ-987165    Strategy 3
## 108 XYZ-987166    Strategy 2
## 109 XYZ-987168    Strategy 1
## 110 XYZ-987170   Strategy NA
## 111 XYZ-987172    Strategy 3
## 112 XYZ-987173    Strategy 2
## 113 XYZ-987174    Strategy 2
## 114 XYZ-987175    Strategy 2
## 115 XYZ-987177    Strategy 1
## 116 XYZ-987179    Strategy 1
## 117 XYZ-987180    Strategy 2
## 118 XYZ-987181    Strategy 2
## 119 XYZ-987183    Strategy 2
## 120 XYZ-987185    Strategy 1
## 121 XYZ-987186    Strategy 2
## 122 XYZ-987187   Strategy NA
## 123 XYZ-987189    Strategy 1
## 124 XYZ-987190    Strategy 2
```

Distribution of best strategy

```
table(bs_result_data_frame$`Best strategy`)
```

```
##
##  Strategy 1  Strategy 2  Strategy 3 Strategy NA
##          32          67          11          14
```

Characteristics of accounts that are influenced most by strategy 1

```
s_accounts = function(strategy) {

  s1_accounts = bs_result_data_frame[bs_result_data_frame$`Best strategy` == strategy, ]

  s1_accounts$acc_type = NA
  s1_accounts$acc_size = NA
  s1_accounts$avg_sales = NA

  for (r in 1:nrow(s1_accounts) ) {

  s1_accounts$acc_type[r] = unique(raw_sales_data$accType[raw_sales_data$accID == s1_accounts$ac
cID[r] ])
  s1_accounts$acc_size[r] = mean(raw_sales_data$accSize[raw_sales_data$accID == s1_accounts$accI
D[r] ])
  s1_accounts$avg_sales[r] = mean(raw_sales_data$sales[raw_sales_data$accID == s1_accounts$accID
[r] ])
  }

  return(s1_accounts)

}

s1_accounts = s_accounts("Strategy 1")
s1_accounts
```

```
##          accID Best strategy       acc_type acc_size   avg_sales
## 1   XYZ-987002    Strategy 1       Pharmacy      700  6641345.42
## 2   XYZ-987005    Strategy 1       Hospital      790 15324815.71
## 6   XYZ-987009    Strategy 1       Pharmacy      520  3196369.46
## 9   XYZ-987015    Strategy 1       Hospital      210  4865584.71
## 10  XYZ-987016    Strategy 1       Hospital      200   248606.46
## 17  XYZ-987025    Strategy 1       Hospital      790 11613475.54
## 18  XYZ-987026    Strategy 1       Hospital      350  2113155.38
## 21  XYZ-987030    Strategy 1 Private Clinic       50   461697.75
## 22  XYZ-987032    Strategy 1 Private Clinic       30    35515.21
## 23  XYZ-987033    Strategy 1       Hospital       50  1349578.21
## 24  XYZ-987034    Strategy 1       Hospital       30    35515.21
## 30  XYZ-987045    Strategy 1       Hospital      370   461697.75
## 32  XYZ-987049    Strategy 1     Polyclinic      579  2379519.46
## 35  XYZ-987054    Strategy 1       Hospital      310  3764612.92
## 39  XYZ-987061    Strategy 1     Polyclinic        0   106545.62
## 50  XYZ-987079    Strategy 1     Polyclinic        0   106545.62
## 52  XYZ-987082    Strategy 1     Polyclinic      190   213091.29
## 64  XYZ-987096    Strategy 1       Hospital      280  1118729.21
## 66  XYZ-987100    Strategy 1       Hospital      400  1314062.79
## 67  XYZ-987102    Strategy 1     Polyclinic       70   159818.46
## 87  XYZ-987137    Strategy 1       Hospital      200  1367335.83
## 88  XYZ-987138    Strategy 1       Pharmacy     1648 22800768.54
## 90  XYZ-987141    Strategy 1       Hospital      610  4581462.79
## 91  XYZ-987143    Strategy 1       Pharmacy      830   198885.33
## 94  XYZ-987146    Strategy 1       Hospital      350  7586050.08
## 103 XYZ-987160    Strategy 1       Pharmacy      240  7103043.08
## 104 XYZ-987161    Strategy 1     Polyclinic      200  1846791.25
## 109 XYZ-987168    Strategy 1       Hospital      718 21575493.62
## 115 XYZ-987177    Strategy 1       Hospital      247  1136486.92
## 116 XYZ-987179    Strategy 1       Hospital      640  1104523.21
## 120 XYZ-987185    Strategy 1       Hospital      330  4652493.25
## 123 XYZ-987189    Strategy 1       Hospital      400 18762688.75
```

Characteristics of accounts that are influenced most by strategy 2

```
s2_accounts = s_accounts("Strategy 2")
s2_accounts
```

```
##          accID Best strategy       acc_type acc_size   avg_sales
## 3   XYZ-987006    Strategy 2       Hospital      320  4350613.92
## 4   XYZ-987007    Strategy 2       Pharmacy     1000 16194938.46
## 5   XYZ-987008    Strategy 2       Hospital      440 13664479.25
## 7   XYZ-987010    Strategy 2       Hospital      670  5860010.67
## 11  XYZ-987017    Strategy 2       Hospital      680 13730182.54
## 12  XYZ-987019    Strategy 2       Hospital       60  3054308.62
## 14  XYZ-987022    Strategy 2       Hospital      420   142060.83
## 15  XYZ-987023    Strategy 2       Pharmacy      720  4563705.38
## 16  XYZ-987024    Strategy 2 Private Clinic      520  3196369.33
## 19  XYZ-987027    Strategy 2       Hospital      380  8772258.33
## 20  XYZ-987029    Strategy 2       Hospital      300  1988852.12
## 26  XYZ-987036    Strategy 2       Hospital      130  1029941.25
## 28  XYZ-987039    Strategy 2       Hospital       30   497212.96
## 29  XYZ-987043    Strategy 2       Hospital      640  9482562.58
## 33  XYZ-987050    Strategy 2 Private Clinic      250  9304986.54
## 36  XYZ-987056    Strategy 2     Polyclinic       20   106545.62
## 37  XYZ-987059    Strategy 2       Hospital       30    71030.42
## 40  XYZ-987062    Strategy 2 Private Clinic       12   266364.12
## 41  XYZ-987063    Strategy 2     Polyclinic      150   994426.12
## 42  XYZ-987065    Strategy 2     Polyclinic       20   177576.04
## 43  XYZ-987067    Strategy 2       Pharmacy      650 10476988.67
## 44  XYZ-987069    Strategy 2     Polyclinic      140  1207517.33
## 45  XYZ-987070    Strategy 2     Polyclinic       30    35515.21
## 47  XYZ-987072    Strategy 2       Pharmacy     1760 15804271.08
## 48  XYZ-987075    Strategy 2     Polyclinic       30   142060.83
## 49  XYZ-987077    Strategy 2       Pharmacy     1770 21451190.38
## 51  XYZ-987081    Strategy 2       Hospital       30    71030.42
## 53  XYZ-987083    Strategy 2     Polyclinic      200   213091.33
## 54  XYZ-987085    Strategy 2       Hospital      287  2201943.38
## 55  XYZ-987086    Strategy 2 Private Clinic        0    35515.21
## 56  XYZ-987087    Strategy 2     Polyclinic       30    35515.21
## 57  XYZ-987088    Strategy 2       Hospital      440 12053864.29
## 58  XYZ-987089    Strategy 2     Polyclinic       20   142060.83
## 59  XYZ-987090    Strategy 2       Pharmacy     1270  9518077.83
## 61  XYZ-987092    Strategy 2       Hospital      660  3480491.17
## 65  XYZ-987097    Strategy 2       Pharmacy      920  2850096.04
## 68  XYZ-987103    Strategy 2       Hospital      500 11222808.17
## 69  XYZ-987104    Strategy 2     Polyclinic      100   106545.62
## 71  XYZ-987107    Strategy 2       Pharmacy      880 12561731.83
## 72  XYZ-987109    Strategy 2     Polyclinic       40    71030.42
## 73  XYZ-987111    Strategy 2       Hospital      260  4581462.88
## 76  XYZ-987118    Strategy 2     Polyclinic       60    35515.21
## 77  XYZ-987119    Strategy 2     Polyclinic      100   177576.04
## 78  XYZ-987120    Strategy 2     Polyclinic       40  2095397.75
## 81  XYZ-987127    Strategy 2     Polyclinic      200   284121.67
## 82  XYZ-987130    Strategy 2       Hospital      300  6996497.50
## 83  XYZ-987133    Strategy 2       Pharmacy      240   994425.96
## 84  XYZ-987134    Strategy 2       Hospital      530 13992995.04
## 86  XYZ-987136    Strategy 2       Hospital      570 13424751.58
## 89  XYZ-987139    Strategy 2       Hospital      275  3018793.33
## 92  XYZ-987144    Strategy 2       Hospital      150  1189759.58
```

```
## 93  XYZ-987145    Strategy 2     Polyclinic       80   177576.04
## 95  XYZ-987147    Strategy 2       Hospital      530  3871158.58
## 96  XYZ-987148    Strategy 2     Polyclinic       90   497212.96
## 97  XYZ-987149    Strategy 2       Pharmacy       20   355152.08
## 98  XYZ-987150    Strategy 2       Hospital      240  1207517.38
## 99  XYZ-987151    Strategy 2     Polyclinic      348   603758.67
## 106 XYZ-987164    Strategy 2       Pharmacy     1710 17393577.08
## 108 XYZ-987166    Strategy 2       Pharmacy      780  7298376.92
## 112 XYZ-987173    Strategy 2       Hospital      590  7103043.04
## 113 XYZ-987174    Strategy 2       Hospital       80  5682434.50
## 114 XYZ-987175    Strategy 2     Polyclinic       20   177576.04
## 117 XYZ-987180    Strategy 2       Hospital      120   177576.04
## 118 XYZ-987181    Strategy 2       Hospital      160        0.00
## 119 XYZ-987183    Strategy 2 Private Clinic      120   248606.50
## 121 XYZ-987186    Strategy 2       Hospital      490 10832140.79
## 124 XYZ-987190    Strategy 2     Polyclinic      200   106545.62
```

Characteristics of accounts that are influenced most by strategy 3

```
s3_accounts = s_accounts("Strategy 3")
s3_accounts
```

```
##           accID Best strategy       acc_type acc_size   avg_sales
## 13  XYZ-987021    Strategy 3       Pharmacy      580  5025403.12
## 25  XYZ-987035    Strategy 3       Hospital       60    88788.04
## 27  XYZ-987037    Strategy 3       Hospital      300  2193064.50
## 34  XYZ-987052    Strategy 3       Hospital      200  1136486.79
## 46  XYZ-987071    Strategy 3       Pharmacy     1026 10175109.29
## 80  XYZ-987124    Strategy 3 Private Clinic       50    35515.21
## 85  XYZ-987135    Strategy 3       Hospital      420  8559167.00
## 101 XYZ-987156    Strategy 3       Pharmacy      965 17722092.62
## 105 XYZ-987163    Strategy 3       Hospital      360  1243032.67
## 107 XYZ-987165    Strategy 3       Hospital      350  4199674.25
## 111 XYZ-987172    Strategy 3       Pharmacy     2460 21664281.67
```

Characteristics of accounts that are influenced most by strategy NA

```
s4_accounts = s_accounts("Strategy NA")
s4_accounts
```

```
##          accID Best strategy       acc_type acc_size avg_sales
## 8   XYZ-987014   Strategy NA     Polyclinic      240 213091.29
## 31  XYZ-987047   Strategy NA Private Clinic       10  71030.42
## 38  XYZ-987060   Strategy NA     Polyclinic       60  71030.42
## 60  XYZ-987091   Strategy NA     Polyclinic      180 781334.58
## 62  XYZ-987093   Strategy NA     Polyclinic       80 355152.08
## 63  XYZ-987095   Strategy NA     Polyclinic       60  71030.42
## 70  XYZ-987106   Strategy NA     Polyclinic       30 461697.71
## 74  XYZ-987112   Strategy NA Private Clinic       50  71030.42
## 75  XYZ-987113   Strategy NA Private Clinic       10 106545.62
## 79  XYZ-987121   Strategy NA     Polyclinic       10  71030.42
## 100 XYZ-987153   Strategy NA     Polyclinic       60  71030.42
## 102 XYZ-987157   Strategy NA       Hospital      350 728061.83
## 110 XYZ-987170   Strategy NA       Hospital       50 106545.62
## 122 XYZ-987187   Strategy NA     Polyclinic       20  35515.21
```

Matrix of best strategy

```
acc_type_s1 = table(s1_accounts$acc_type)
acc_type_s2 = table(s2_accounts$acc_type)
acc_type_s3 = table(s3_accounts$acc_type)
acc_type_s4 = table(s4_accounts$acc_type)

acc_matrix = matrix(0, nrow = 4, ncol = 4)
colnames(acc_matrix) = c("Hospital", "Pharmacy", "Polyclinic", "Private Clinic")
rownames(acc_matrix) = c("Strategy 1", "Strategy 2", "Strategy 3", "Strategy NA")

acc_matrix["Strategy 1", ]  = acc_type_s1[c("Hospital", "Pharmacy", "Polyclinic", "Private Clini
c")]
acc_matrix["Strategy 2", ]  = acc_type_s2[c("Hospital", "Pharmacy", "Polyclinic", "Private Clini
c")]
acc_matrix["Strategy 3", ]  = acc_type_s3[c("Hospital", "Pharmacy", "Polyclinic", "Private Clini
c")]
acc_matrix["Strategy NA", ] = acc_type_s4[c("Hospital", "Pharmacy", "Polyclinic", "Private Clini
c")]

acc_matrix[is.na(acc_matrix) ] = 0
acc_matrix
```

```
##             Hospital Pharmacy Polyclinic Private Clinic
## Strategy 1        19        5          6              2
## Strategy 2        30       12         20              5
## Strategy 3         6        4          0              1
## Strategy NA        2        0          9              3
```

Matrix of percentage of best strategy

```
acc_matrix_percent = acc_matrix

for (c in 1:ncol(acc_matrix) ) {
  acc_matrix_percent[, c] = acc_matrix[, c] / sum(acc_matrix[, c] )
}
acc_matrix_percent
```

```
##               Hospital  Pharmacy Polyclinic Private Clinic
## Strategy 1  0.33333333 0.2380952  0.1714286     0.18181818
## Strategy 2  0.52631579 0.5714286  0.5714286     0.45454545
## Strategy 3  0.10526316 0.1904762  0.0000000     0.09090909
## Strategy NA 0.03508772 0.0000000  0.2571429     0.27272727
```

```
library("formattable")

percent(acc_matrix_percent)
```

```
##             Hospital Pharmacy Polyclinic Private Clinic
## Strategy 1  33.33%   23.81%   17.14%     18.18%
## Strategy 2  52.63%   57.14%   57.14%     45.45%
## Strategy 3  10.53%   19.05%   0.00%      9.09%
## Strategy NA 3.51%    0.00%    25.71%     27.27%
```