3/22/2017

B.Sc in Computing
Enterprise Database Technologies

David Lawlor X00107563

Contents

Introduction

The aim of the assignment is to analyse a data set. The data set which we were given was a telecommunication provider which provides fixed line, mobile and broadband services to its customers.

The aim of the analysis is to try to discover why customers regularly change service providers or churn. The analysis will provide insight into why customers churn, enabling them to identify gaps in their service offering and increase their overall customer retention.

The analysis will use the CRISP-DM process

- data understanding,
- data pre-processing stages
- Data Mining task.

Q1 Data Understanding

The dataset contained 2071 rows containing 18 columns.

To allow the building of an accurate model it is imperative that the dataset is cleaned by addressing duplicates, missing values and other erroneous data.

For each predictor variable, 16 in total, a table was created to detail the various characteristics of the predictor. In all cases these were data type (nominal, ordinal, numerical), percentage of missing values and mode. For the numerical data, the min, max, mean, median and standard deviation was also included. (Appendix 1.3)

Missing Data

One step taken to understand the data was to print out a summary of the data. This enabled the identification of columns with NA values or blank values. The following table outlines the rows found to have missing values. (Appendix 1.1.2)

Column	Number of missing values
MINUTES_3MONTHS_AGO	3 Records
CUST_MOS	3 Records
TOT_MINUTES_USAGE	4 Records
PHONE_PLAN	4 Records
EDUCATION	8 Records

For the numeric attributes, MINUTES_3MONTHS_AGO, TOT_MINUTES_USAGE and CUST_MOS they were replaced with the median value for the columns.

For the EDUCATION and PHONE_PLAN the missing values were replaced with the mode of the column based on the GENDER column e.g.: for Education, if the record was male, the missing value was replaced with the mode of education for that gender. (Appendix 1.1.2)

Duplicate Data

There was 1 duplicate row identified. This row was positively identified as a duplicate as the CUST_ID was duplicated. This row was removed so that it did not impact the overall analysis. While there were other rows which were observed to be identical, exclusion was not possible due to the records having different CUST_ID's (Appendix 1.1.1.1)

Other data quality issues

A data quality issue which was observed was in the TOT_MINUTES_USAGE. This Column is the sum of MINUTES_3MONTHS_AGO, MINUTES_PREV_MONTH AND MINUTES_CURR_MONTH. Although this is the case for much of the dataset, there were 184 rows in which the TOT_MINUTES_USAGE was zero despite the other 3 columns in the row having non-zero data.

Q2 Discretise income predictor

For the income, it was discretized using a binning operation which grouped income into an ordinal predictor consisting of Low Income (< 38,000), Medium Income (>= 38,000 and < 88,000) and High Income (>= 88,000). (Appendix 1.2)

Q3 Predictor information

For the numerical predictors in the dataset all had a positive skewness when calculated with no transformations applied to the data. The MINUTES_3MONTHS_AGO predictor had the lowest positive skewness of 0.901 and NUM_LINES having the highest positive skewness. This means that all the numerical predictors do not have a normal distribution. Each predictor's full characteristics can be found in full detain in Appendix 1.3.

The numerical predictors are covered in Q6.

Q4 Identifying Outliers

The numeric predictor chosen for outliers is TOT_MINUTES_USAGE. This is the total minutes' usage for a customer based on 3 months of minutes. (Appendix 1.4)

IQR method

The interquartile method identified 0 outliers in the lower range and 176 in the upper range. This was interesting as it meant that 8.5% of the entire dataset were outliers in the upper range

z-Score Standardisation method

Apply a z-score transformation on the column identified again 0 outliers in the lower range and just 69 in the upper range compare to 176 using the IQR method.

Q5 Skewness

The numerical variable chosen for testing for skewness was TOT_MINUTES_USAGE. As a before any transformation was applied to the predictor the skewness measurement was taken. The predictor has a positive skewness of 1.089144. Three transformation methods were attempted to try to correct this skewness. (Appendix 1.5)

Natural Log Transformation

The natural log transformation transformed the data from a positive skewness to a negative skewness of -0.7042918. This method also required the exclusion of zero values from the calculation.

Square Root Transformation

The square root transformation had a negative impact on the skewness by increasing the positive skewness of the data to 1.289714 so this eliminated this transformation as a possibility.

z-Score Standardisation

The z-Score standardisation had no effect on the skewness as would be expected to be the case as z-Score does not actually transform the data but standardises it.

Q6 Numerical Predictor with Response Variable Overlay

The numerical predictors with the churn overlay did not show anything significant within the minute's data in terms of a pattern of even a consistent split between churners.

The CUST_MOS did show that customers which were with the company more than 44 months has a 100% churn rate.

The NUM_LINES did not appear to show and value to determining churn as each value had a nearly even split

Q7 Investigating correlated variable

To investigate the correlation between the numerical variables they were each compared to one another graphically using a scatter plot.

All the numerical variables concerning minutes used show a correlation with each other. As the total minutes used is the sum of 3 months previous, previous month and current month, this was expected as they are all related to one another. When the minute columns were then compared to the other two numerical variables customer months and number of lines there was no correlation.

These correlations were also performed to mathematically to affirm these results. (Appendix 1.7)

For all the predictors, not all showed a possible reason for churn. Some of the main ones which did show a possible influence in the churn rate were:

Predictor	Possible reason for churn
AREA_CODE	from the 6 area codes in the dataset, one area 21750 had a churn rate of 80%.
INCOME	The churn for medium income is much higher when compared to the churn rate for the low and medium incomes.
PHONE_PLAN	While there is not a huge number of customers (59) on the Euro-Zone plan in the dataset it does indicate a high churn of 85%. The international plan also has a high churn of 64%
EDUCATION	Primary level education has a 100% churn rate of the 91 record in the dataset. Masters education also has a high churn of 67%.
CUST_MOS	The churn is 100% after 44 months.

Weka

For the data mining task the data mining software Weka was used to create a model of the data. Several different algorithms are used as each create their own model which can then be analysed.

The output of the various algorithms is summarised in the table below. The formulas used to calculate these are detailed in (Appendix 2)

	J48	JRip	PART	ZeroR
Proportion of false positives	0.044	0.041	0.071	0.479
Proportion of false negatives	0.282	0.289	0.271	0
Overall error rate	0.197	0.203	0.195	0.479
Overall Model Accuracy	0.803	0.797	0.805	0.521
Precision	0.956	0.959	0.929	0.521
Sensitivity(Recall) - True Positive Rate	0.651	0.638	0.678	1
Specificity - False Positive Rate	0.967	0.97	0.944	0
ROC	0.843	0.809	0.847	0.5

For the analysis, the predictors CUST_ID, MINUTES_CURR_MONTH, MINUTES_PREV_MONTH, MINUTES_3MONTHS_AGO were excluded from the analysis. This was done from within the Weka application using the filters.

The unsupervised attribute filter NumericToNominal was applied to the AREA_CODE, LONGDIST_FLAG, CALLWAITING_FLAG, VOICEMAIL_FLAG and MOBILE_PLAN. This was required as these nominal predictors were imported as numerical. The normalise filter was also applied to the numerical data.

ZeroR

Description of Algorithm

Class for building and using a 0-R classifier. Predicts the mean (for a numeric class) or the mode (for a nominal class). The algorithm looks at the yes/no churn split and bases its guessed on this.

As expected in this best guess algorithm the results show a rough 50 50 split on classifying as there is a near 50/50 split on the churn rate with 1020 No and 1051 yes.

This is also backed up by the Roc curve of 0.5 which show it has a 50/50 of predicting a churner

JRip

Description of Algorithm

The JRip algorithm is a java implementation of the Ripper machine learning algorithm.

The predictor used in generating the model were:

- CONVERGENT_BILLING
- TOT_MINUTES_USAGE
- AREA_CODE
- CUST_MOS
- EDUCATION
- INCOME

Although didn't perform quite as well as J48 and PART. The algorithm used just 6 predictors in the model and produced just 7 rules in total.

The algorithm used a combination of CONVERGENT_BILLING and TOT_MINUTES_USAGE to classify most of the record with 690 out of 890 correctly classified.

J48

Description of Algorithm

J48 is a decision tree divide and conquer algorithm which is the Weka implementation of Iterative Dichotomiser 3(ID3). ID3 is a precursor to C4.5. Splits are based on the maximum information gain.

The predictor used in generating the model were:

- CUST_MOS
- CONVERGENT_BILLING
- GENDER
- AREA_CODE

- INCOME
- TOT_MINUTES_USAGE
- EDUCATION
- VOICEMAIL_FLAG
- PHONE_PLAN

This means that the algorithm used a total of 9/13 of the possible predictors in creating the model.

This algorithm identified the 100% churn after 44 months and used it as the first split in the tree. Interestingly after a split on CONVERGENT_BILLING, EDUCATION is only used in the no side of the tree for further splits.

PART

Description of Algorithm

The PART algorithm creates a s partial j48 decision tree and makes the best leaf into a rule.

The predictor used in generating the model were:

- CUST_MOS
- CONVERGENT_BILLING
- GENDER
- AREA_CODE
- INCOME
- TOT_MINUTES_USAGE
- EDUCATION
- VOICEMAIL_FLAG
- PHONE_PLAN

The false positive rate is the lowest and its true positive rate is the highest. Similarly, J48 used the same 9 predictors in the model.

Comparing this to the JRip algorithm, PART produced a total of 17 rules compare to just 7 in JRip. The algorithm used a combination of PHONE_PLAN and INCOME to correctly classify 270 out of 350 records.

Weka Conclusion

If based on solely on the ROC curve, the PART algorithm preforms the best at predicting churn with an 84.7% chance of predicting churn but only marginally with J48 having an 84.3% prediction rate.

Surprisingly, convergent billing is used in all models despite not showing a lot of potential in the histogram with churn overlay (Appendix 1.3.6).

The algorithms used the predictors which were identified from the histograms earlier apart from JRip which did not use phone plan.

Conclusion

In conclusion, there seems to be no one single predictor of churn but more of a combination of several.

CUST_MOS would be one of the reasons for churn. A 100% churn after 44 months combined with the fact that it is used in all algorithms. In PART, it is used by itself as a single rule and in j48 it is used as the first split in the tree which means it immediately produced a pure node in the tree. As good as this predictor is for predicting churn over 44 months, less than that its predictive value decreases.

EDUCATION is a factor. Primary level education has a 100% churn rate and masters level education has a high churn rate of 69%. Looking at this when it is used it is used in J48, this seems to have more of an importance when customer months are less than 44 and convergent billing is NO.

AREA_CODE is a good predictor for churn. Its presence is prevalent across all models. This was evident from the histogram before the mining stage with Weka. This is an indication that some areas are more likely to churn which could be due to poor services or lack of advertising in these area compared to their competitors.

Appendix

REFERENCES: http://stackoverflow.com/questions/1330989/rotating-and-spacing-axis-labels-in-ggplot2
http://stackoverflow.com/questions/30057765/histogram-ggplot-show-count-label-for-each-bin-for-each-category

---

```
library(ggplot2)
```

Functions

The following function is the function which will be used to get the mode of the data

```
mymode <- function(x){
  xtable <- table(x)
  idx <- xtable == max(xtable)
  names(xtable)[idx]
}
```

Read in the data from the csv file

```
churn_data <- read.csv("./eurocomPHONEchurners.csv")
```

First thing to do is to print out the data summary.

```
summary(churn_data)

##     CUST_ID         AREA_CODE     MINUTES_CURR_MONTH MINUTES_PREV_MONTH
##  Min.   :   1.0   Min.   :10040   Min.   :    1.0    Min.   :    0.0
##  1st Qu.: 517.5   1st Qu.:15563   1st Qu.:   33.0    1st Qu.:   15.0
##  Median :1035.0   Median :21750   Median :  105.0    Median :   98.0
##  Mean   :1035.1   Mean   :29816   Mean   :  747.7    Mean   :  863.9
##  3rd Qu.:1552.5   3rd Qu.:45987   3rd Qu.:  555.0    3rd Qu.:  444.0
##  Max.   :2070.0   Max.   :55166   Max.   :14000.0    Max.   :16754.0

##  MINUTES_3MONTHS_AGO    CUST_MOS     LONGDIST_FLAG    CALLWAITING_FLAG
##  Min.   :    0       Min.   : 1.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:   29       1st Qu.: 6.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :   97       Median :11.00   Median :1.0000   Median :0.0000
##  Mean   :  453       Mean   :16.05   Mean   :0.5649   Mean   :0.4346
##  3rd Qu.:  598       3rd Qu.:26.00   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :12456       Max.   :50.00   Max.   :1.0000   Max.   :1.0000
##  NA's   :3           NA's   :3
```

```
##    NUM_LINES      VOICEMAIL_FLAG    MOBILE_PLAN     CONVERGENT_BILLING
##  Min.   :1.000   Min.   :0.0000   Min.   :0.0000   No :1241
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.0000   Yes: 830
##  Median :1.000   Median :1.0000   Median :0.0000
##  Mean   :1.391   Mean   :0.5654   Mean   :0.3477
##  3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.0000   Max.   :1.0000

##  GENDER        INCOME                  PHONE_PLAN            EDITUCATION
##  F: 731   Min.   : 17000                  :   4                 :  8
##  M:1340   1st Qu.: 46000   Euro-Zone     :  59   Bachelors    :400
##           Median : 75000   International:1067   High School :  3
##           Mean   : 85784   National      : 671   Masters      :330
##           3rd Qu.: 98000   Promo_plan    : 270   PhD          :410
##           Max.   :320000                         Post Primary:829
                                                     Primary      : 91


##  TOT_MINUTES_USAGE CHURNER
##  Min.   :    0     no :1021
##  1st Qu.:  116     yes:1050
##  Median :  264
##  Mean   : 2040
##  3rd Qu.: 1677
##  Max.   :36237
##  NA's   :4
```

1.1 Preprocess the data

1.1.1.1 Identify duplicate values

ref: http://www.cookbook-r.com/Manipulating_data/Finding_and_removing_duplicate_records/

```
churn_data[duplicated(churn_data$CUST_ID), ]

##     CUST_ID AREA_CODE MINUTES_CURR_MONTH MINUTES_PREV_MONTH
## 152     246     10040                  2                  0
##     MINUTES_3MONTHS_AGO CUST_MOS LONGDIST_FLAG CALLWAITING_FLAG NUM_LINES
## 152                   0        1             0                0         1
##     VOICEMAIL_FLAG MOBILE_PLAN CONVERGENT_BILLING GENDER INCOME PHONE_PLAN
## 152              1           0                 No      F  29000   National
##        EDUCATION TOT_MINUTES_USAGE CHURNER
## 152 Post Primary                 0      no
```

1.1.1.2 Remove the duplicates

```
churn_data <- churn_data[!duplicated(churn_data$CUST_ID), ]

nrow(churn_data)

## [1] 2070
```

1.1.1.3 Identify null values

The summary data can be used to identify the columns which contain null values.

The columns with null values are:

- MINUTES_3MONTHS_AGO (3 records)
- CUST_MOS (3 records)
- TOT_MINUTES_USAGE (4 records)
- PHONE_PLAN (4 records)
- EDUCATION (8 records)

1.1.2 Addressing the issues of the missing values.

For the numeric values they will be replaced with median value for the columns

1.1.2.1 MINUTES_3MONTHS_AGO

```
median(churn_data$MINUTES_3MONTHS_AGO, na.rm = TRUE)

## [1] 97

MINUTES_3MONTHS_AGO_median <- median(churn_data$MINUTES_3MONTHS_AGO, na.rm =
TRUE)

churn_data$MINUTES_3MONTHS_AGO[is.na(churn_data$MINUTES_3MONTHS_AGO)] <-
MINUTES_3MONTHS_AGO_median
```

1.1.2.2 CUST_MOS

```
median(churn_data$CUST_MOS, na.rm = TRUE)

## [1] 11

CUST_MOS_median <- median(churn_data$CUST_MOS, na.rm = TRUE)

churn_data$CUST_MOS[is.na(churn_data$CUST_MOS)] <- CUST_MOS_median
```

1.1.2.3 TOT_MINUTES_USAGE

```
median(churn_data$TOT_MINUTES_USAGE, na.rm = TRUE)

## [1] 264
```

```
TOT_MINUTES_USAGE_median <- median(churn_data$TOT_MINUTES_USAGE, na.rm = TRUE)

churn_data$TOT_MINUTES_USAGE[is.na(churn_data$TOT_MINUTES_USAGE)] <-
TOT_MINUTES_USAGE_median
```

1.1.2.4 PHONE_PLAN

The first thing to do was to create a data frame which is comprised of the gender and phone plan

```
  phone_plan_gender_subset <- data.frame(churn_data$PHONE_PLAN,
churn_data$GENDER)
```

The data frame is then filtered again so it just contains data for record for male records

```
  phone_plan_male_subset <-
phone_plan_gender_subset[phone_plan_gender_subset$churn_data.GENDER == 'M', 1]

  phone_plan_male_subset[1:10]

##  [1] International International International Promo_plan    National
##  [6] National      International National      National      International
## Levels:  Euro-Zone International National Promo_plan
```

The mode of the phone plan is stored for male records

```
  print(phone_plan_male_mode <- mymode(phone_plan_male_subset))

## [1] "International"
```

The data frame is then filtered again so it just contains data for record for female records

```
  phone_plan_female_subset <-
phone_plan_gender_subset[phone_plan_gender_subset$churn_data.GENDER == 'F', 1]

  phone_plan_female_subset[1:10]

##  [1] National      International International National      International
##  [6] International National      International National      International
## Levels:  Euro-Zone International National Promo_plan

  print(phone_plan_female_mode <- mymode(phone_plan_female_subset))

## [1] "International"
```

We then find the indexes of the phone plan records with empty values

```
  no_phone_plans <- which(churn_data$PHONE_PLAN == "")

for (index in no_phone_plans)
{
```

```
  if (churn_data$GENDER[index] == 'M')
  {
    churn_data$PHONE_PLAN[index] <- phone_plan_male_mode
  }
  else
  {
    churn_data$PHONE_PLAN[index] <- phone_plan_female_mode
  }
}

  churn_data$PHONE_PLAN[no_phone_plans]

## [1] International International International International
## Levels:  Euro-Zone International National Promo_plan

churn_data$PHONE_PLAN <- droplevels(churn_data$PHONE_PLAN)

summary(churn_data$PHONE_PLAN)

##     Euro-Zone International      National    Promo_plan
##            59          1071           670           270
```

1.1.2.5 EDUCATION

The first thing to do was to create a data frame which is comprised of the gender and education

```
  education_gender_subset <- data.frame(churn_data$EDUCATION, churn_data$GENDER)
```

The data frame is then filtered again so it just contains data for record for male records

```
  no_education_male_subset <-
education_gender_subset[education_gender_subset$churn_data.GENDER == 'M', 1]
```

The mode of the phone plan is stored for male records

```
  no_education_male_mode <- mymode(no_education_male_subset)

  no_education_male_mode

## [1] "Post Primary"
```

The data frame is then filtered again so it just contains data for record for female records

```
  no_education_female_subset <-
education_gender_subset[education_gender_subset$churn_data.GENDER == 'F', 1]

  no_education_female_mode <- mymode(no_education_female_subset)

  no_education_female_mode
```

```
## [1] "Bachelors"
```

We then find the indexes of the phone plan records with empty values

```
  no_education <- which(churn_data$EDUCATION == "")

for (index in no_education)
{
  if (churn_data$GENDER[index] == 'M')
  {
    churn_data$EDUCATION[index] <- no_education_male_mode
  }
  else
  {
    churn_data$EDUCATION[index] <- no_education_female_mode
  }
}

churn_data$EDUCATION[no_education]

## [1] Post Primary Bachelors    Post Primary Post Primary Post Primary
## [6] Post Primary Bachelors    Bachelors
## Levels:  Bachelors High School Masters PhD Post Primary Primary

churn_data$EDUCATION <- droplevels(churn_data$EDUCATION)

summary(churn_data)

##     CUST_ID          AREA_CODE     MINUTES_CURR_MONTH MINUTES_PREV_MONTH
##  Min.   :   1.0   Min.   :10040   Min.   :     1.0    Min.   :     0.0
##  1st Qu.: 518.2   1st Qu.:15563   1st Qu.:    33.0    1st Qu.:    15.0
##  Median :1035.5   Median :21750   Median :   105.0    Median :    98.0
##  Mean   :1035.5   Mean   :29826   Mean   :   748.1    Mean   :   864.3
##  3rd Qu.:1552.8   3rd Qu.:45987   3rd Qu.:   555.0    3rd Qu.:   444.0
##  Max.   :2070.0   Max.   :55166   Max.   :14000.0    Max.   :16754.0


##  MINUTES_3MONTHS_AGO    CUST_MOS     LONGDIST_FLAG    CALLWAITING_FLAG
##  Min.   :    0.0     Min.   : 1.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:   29.0     1st Qu.: 6.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :   97.0     Median :11.00   Median :1.0000   Median :0.0000
##  Mean   :  452.7     Mean   :16.05   Mean   :0.5652   Mean   :0.4348
##  3rd Qu.:  598.0     3rd Qu.:26.00   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :12456.0     Max.   :50.00   Max.   :1.0000   Max.   :1.0000



##    NUM_LINES      VOICEMAIL_FLAG    MOBILE_PLAN     CONVERGENT_BILLING
```

```
##  Min.   :1.000   Min.   :0.0000   Min.   :0.0000   No :1240
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.0000   Yes: 830
##  Median :1.000   Median :1.0000   Median :0.0000
##  Mean   :1.391   Mean   :0.5652   Mean   :0.3478
##  3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.0000   Max.   :1.0000

##  GENDER       INCOME                PHONE_PLAN          EDUCATION
##  F: 730    Min.   : 17000    Euro-Zone     :  59    Bachelors    :403
##  M:1340    1st Qu.: 46000    International:1071    High School :  3
##            Median : 75000    National      : 670    Masters      :330
##            Mean   : 85812    Promo_plan    : 270    PhD          :410
##            3rd Qu.: 98000                           Post Primary:833
##            Max.   :320000                           Primary      : 91

##  TOT_MINUTES_USAGE CHURNER
##  Min.   :    0     no :1020
##  1st Qu.:  116     yes:1050
##  Median :  264
##  Mean   : 2037
##  3rd Qu.: 1677
##  Max.   :36237
```

1.2 Discretise Income predictor

Bin the data into categories of low income, medium income and high income.

```
churn_data$INCOME <- cut(churn_data$INCOME, breaks=c(0,38000,88000,Inf), right =
FALSE, labels=c("low income","medium income","high income"))
```

1.3 Predictor Information

1.3.1 AREA_CODE

Geographical area account holder resides

Predictor	AREA_CODE
attribute type	**Nominal**
%Missing Values	**0%**
Mode	**10040, 15563, 21750, 36785, 45987**

CHART CODE

```
plot_data <- data.frame(table(churn_data$AREA_CODE, churn_data$CHURNER))
colnames(plot_data) <- c("Area_Code", "Churned", "Count" )
ggplot(data=plot_data, aes(x = Area_Code, y = Count, fill = Churned, label =
Count)) +
  geom_bar(stat = "identity") +
  geom_text(size = 3, position = position_stack(vjust = 0.5))
```

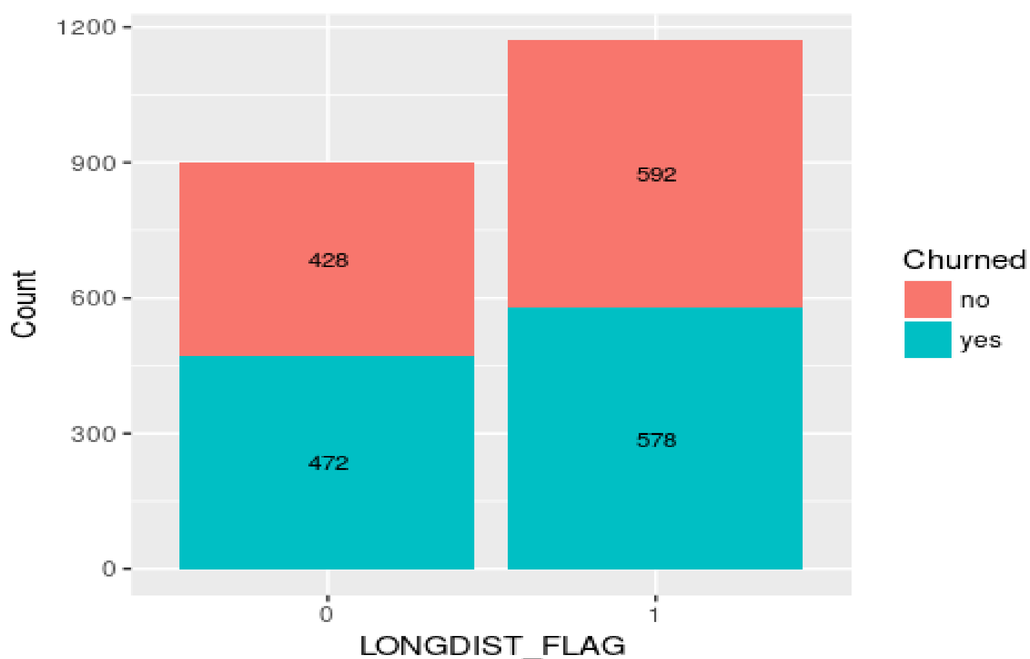
1.3.2 LONGDIST_FLAG

Whether has signed up for the off-peak long distance call package

Predictor	LONGDIST_FLAG
attribute type	**Nominal**
%Missing Values	**0%**
Mode	**1**

CHART CODE

```
plot_data <- data.frame(table(churn_data$LONGDIST_FLAG, churn_data$CHURNER))
colnames(plot_data) <- c("LONGDIST_FLAG", "Churned", "Count" )
ggplot(data=plot_data, aes(x = LONGDIST_FLAG, y = Count, fill = Churned, label =
Count)) +
  geom_bar(stat = "identity") +
  geom_text(size = 3, position = position_stack(vjust = 0.5))
```



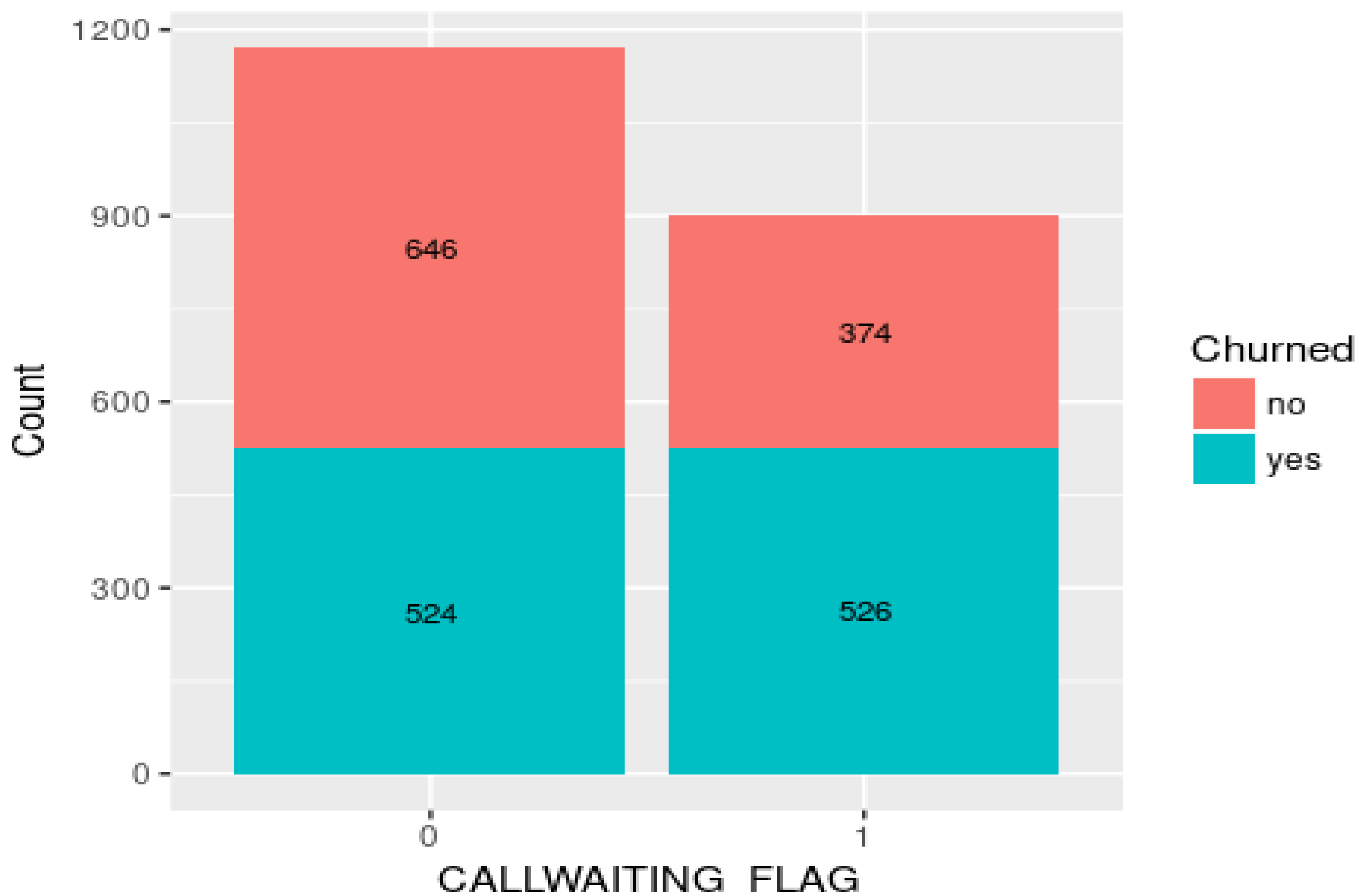
1.3.3 CALLWAITING_FLAG

Whether the customer has call waiting

Predictor	CALLWAITING_FLAG
attribute type	**Nominal**
%Missing Values	**0%**
Mode	**0**

CHART CODE

```
plot_data <- data.frame(table(churn_data$CALLWAITING_FLAG, churn_data$CHURNER))
colnames(plot_data) <- c("CALLWAITING_FLAG", "Churned", "Count")
ggplot(data=plot_data, aes(x = CALLWAITING_FLAG, y = Count, fill = Churned, label
= Count)) +
  geom_bar(stat = "identity") +
  geom_text(size = 3, position = position_stack(vjust = 0.5))
```



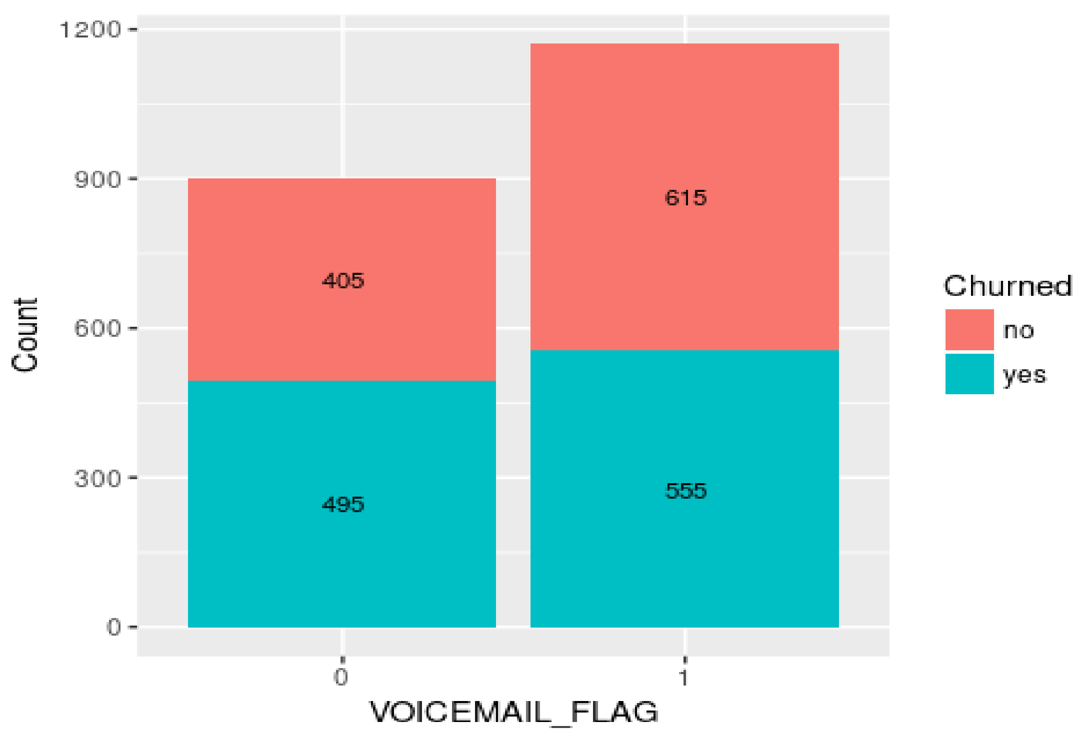
1.3.4 VOICEMAIL_FLAG

Whether the customer has voice mail

Predictor	VOICEMAIL_FLAG
attribute type	**Nominal**
%Missing Values	**0%**
Mode	**1**

CHART CODE

```
plot_data <- data.frame(table(churn_data$VOICEMAIL_FLAG, churn_data$CHURNER))
colnames(plot_data) <- c("VOICEMAIL_FLAG", "Churned", "Count" )
ggplot(data=plot_data, aes(x = VOICEMAIL_FLAG, y = Count, fill = Churned, label =
Count)) +
  geom_bar(stat = "identity") +
  geom_text(size = 3, position = position_stack(vjust = 0.5))
```



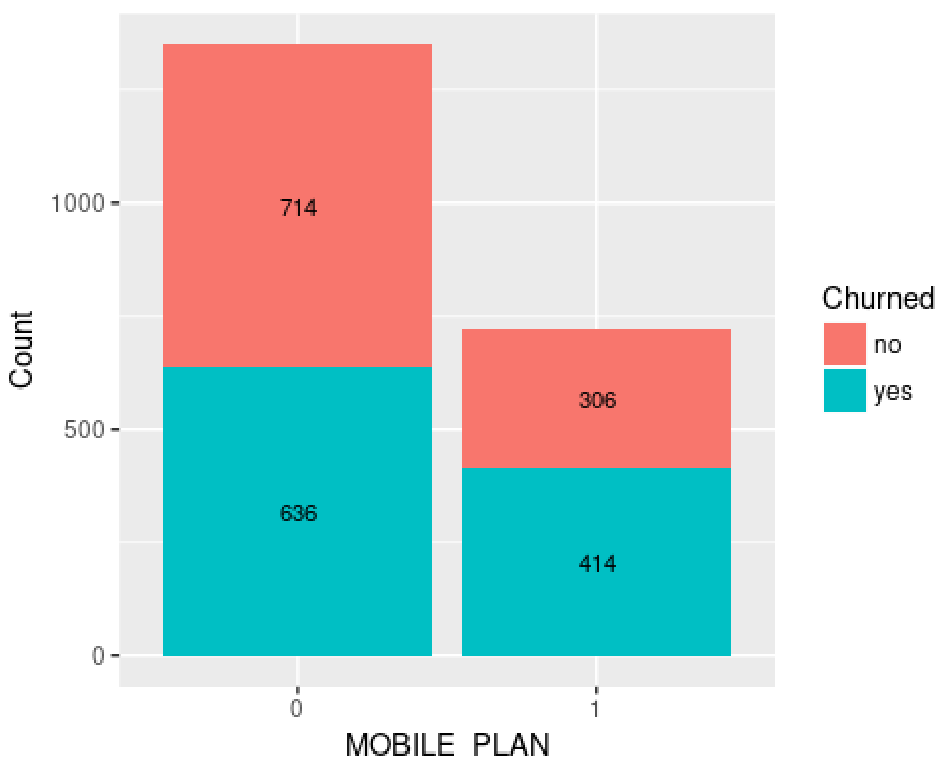
1.3.5 MOBILE_PLAN

Has the customer signed up to the mobile phone plan

Predictor	MOBILE_PLAN
attribute type	**Nominal**
%Missing Values	**0%**
Mode	**0**

CHART CODE

```
plot_data <- data.frame(table(churn_data$MOBILE_PLAN, churn_data$CHURNER))
colnames(plot_data) <- c("MOBILE_PLAN", "Churned", "Count" )
ggplot(data=plot_data, aes(x = MOBILE_PLAN, y = Count, fill = Churned, label =
Count)) +
  geom_bar(stat = "identity") +
  geom_text(size = 3, position = position_stack(vjust = 0.5))
```



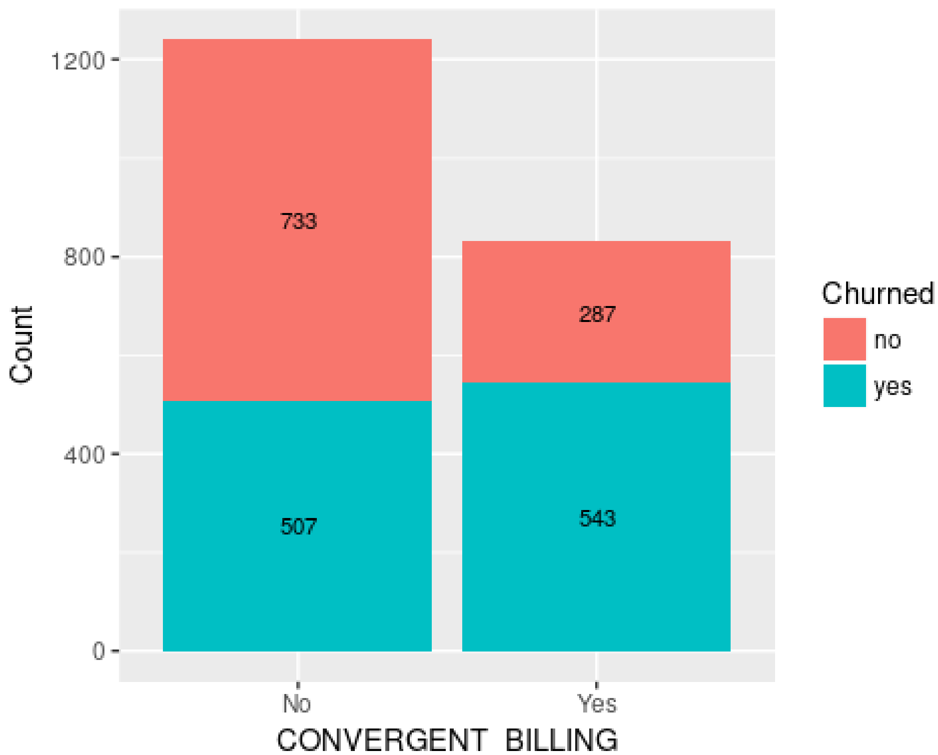
1.3.6 CONVERGENT_BILLING

All service charges consolidated onto one bill

Predictor	CONVERGENT_BILLING
attribute type	**Nominal**
%Missing Values	**0%**
Mode	**No**

CHART CODE

```
plot_data <- data.frame(table(churn_data$CONVERGENT_BILLING, churn_data$CHURNER))
colnames(plot_data) <- c("CONVERGENT_BILLING", "Churned", "Count" )
ggplot(data=plot_data, aes(x = CONVERGENT_BILLING, y = Count, fill = Churned,
label = Count)) +
  geom_bar(stat = "identity") +
  geom_text(size = 3, position = position_stack(vjust = 0.5))
```



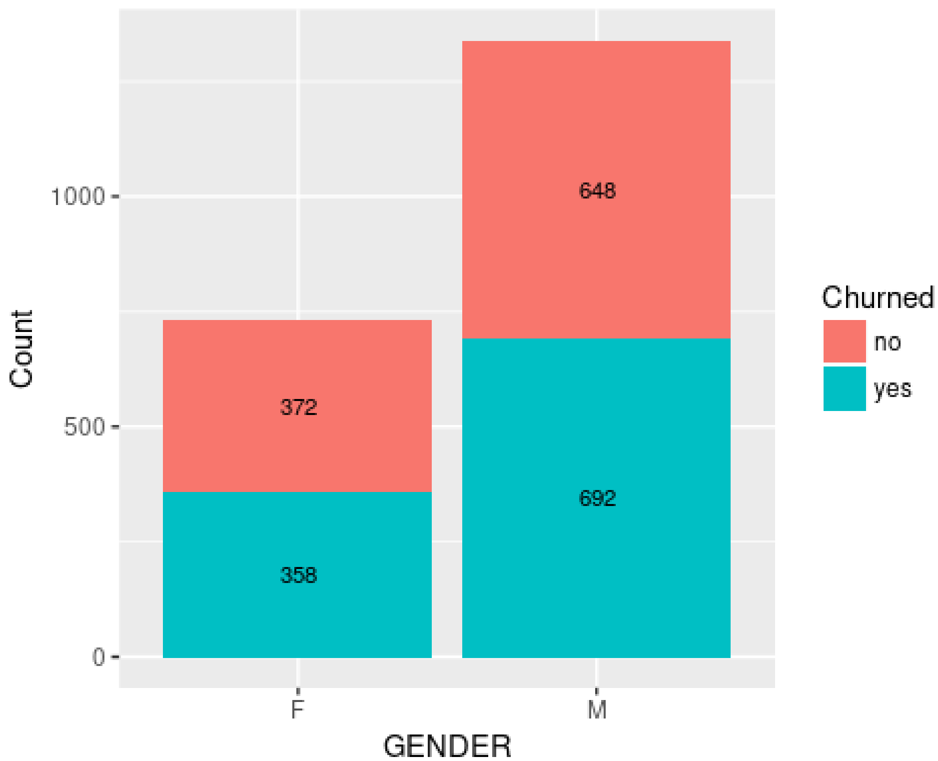
1.3.7 GENDER

Account holder's gender

Predictor	GENDER
attribute type	Nominal
%Missing Values	0%
Mode	M

CHART CODE

```
plot_data <- data.frame(table(churn_data$GENDER, churn_data$CHURNER))
colnames(plot_data) <- c("GENDER", "Churned", "Count" )
ggplot(data=plot_data, aes(x = GENDER, y = Count, fill = Churned, label = Count))
+
  geom_bar(stat = "identity") +
  geom_text(size = 3, position = position_stack(vjust = 0.5))
```



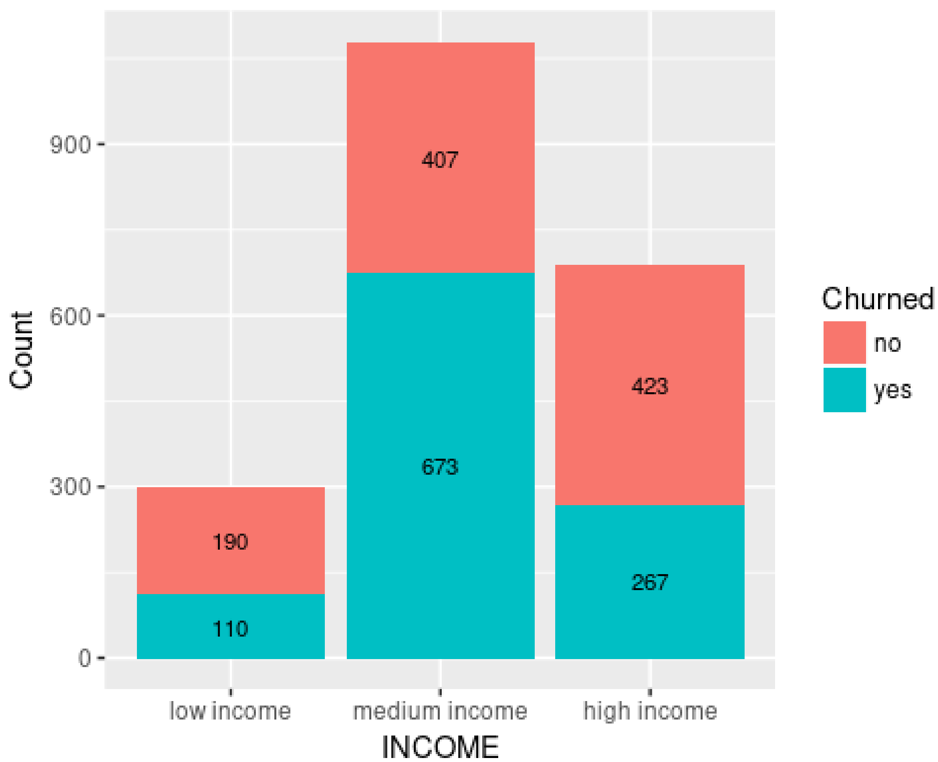
1.3.8 INCOME

Account holder's annual income (€)

Predictor	INCOME
attribute type	Ordinal
%Missing Values	0%
Mode	medium income

CHART CODE

```
plot_data <- data.frame(table(churn_data$INCOME, churn_data$CHURNER))
colnames(plot_data) <- c("INCOME", "Churned", "Count" )
ggplot(data=plot_data, aes(x = INCOME, y = Count, fill = Churned, label = Count))
+
  geom_histogram(stat = "identity") +
  geom_text(size = 3, position = position_stack(vjust = 0.5))

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



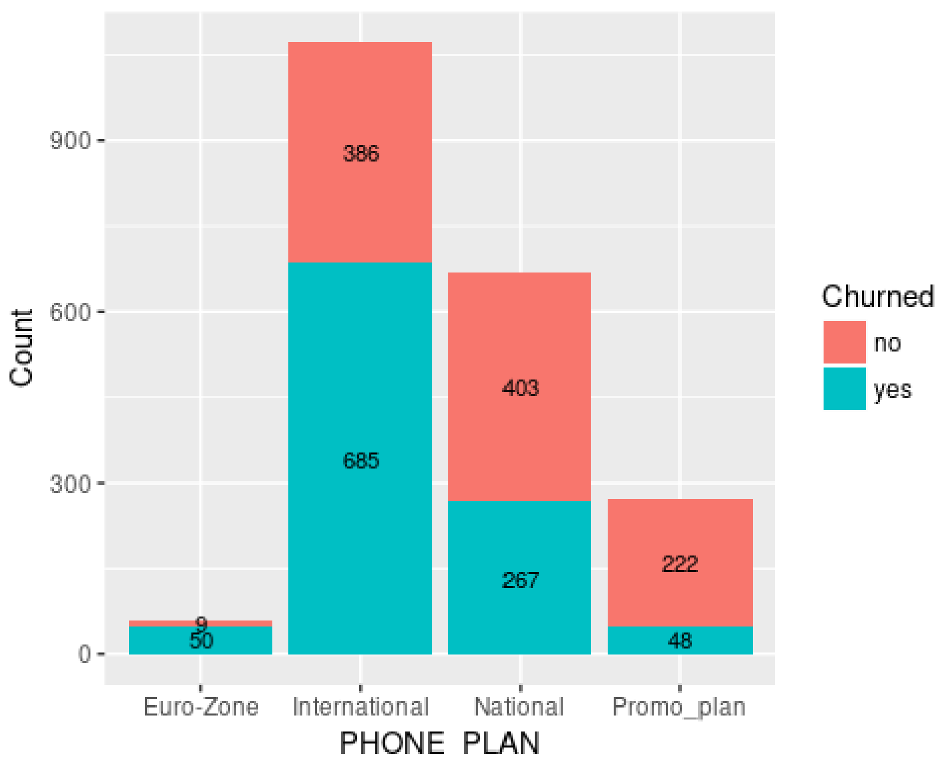
1.3.9 PHONE_PLAN

The phone plan the customer has signed up for national, euro-zone, international (outside Euro-zone) and promo_plan (signed up to the promotional plan)

Predictor	PHONE_PLAN
attribute type	**Nominal**
%Missing Values	**0.193%**
Mode	**International**

CHART CODE

```
plot_data <- data.frame(table(churn_data$PHONE_PLAN, churn_data$CHURNER))
colnames(plot_data) <- c("PHONE_PLAN", "Churned", "Count" )
ggplot(data=plot_data, aes(x = PHONE_PLAN, y = Count, fill = Churned, label =
Count)) +
  geom_bar(stat = "identity") +
  geom_text(size = 3, position = position_stack(vjust = 0.5))
```



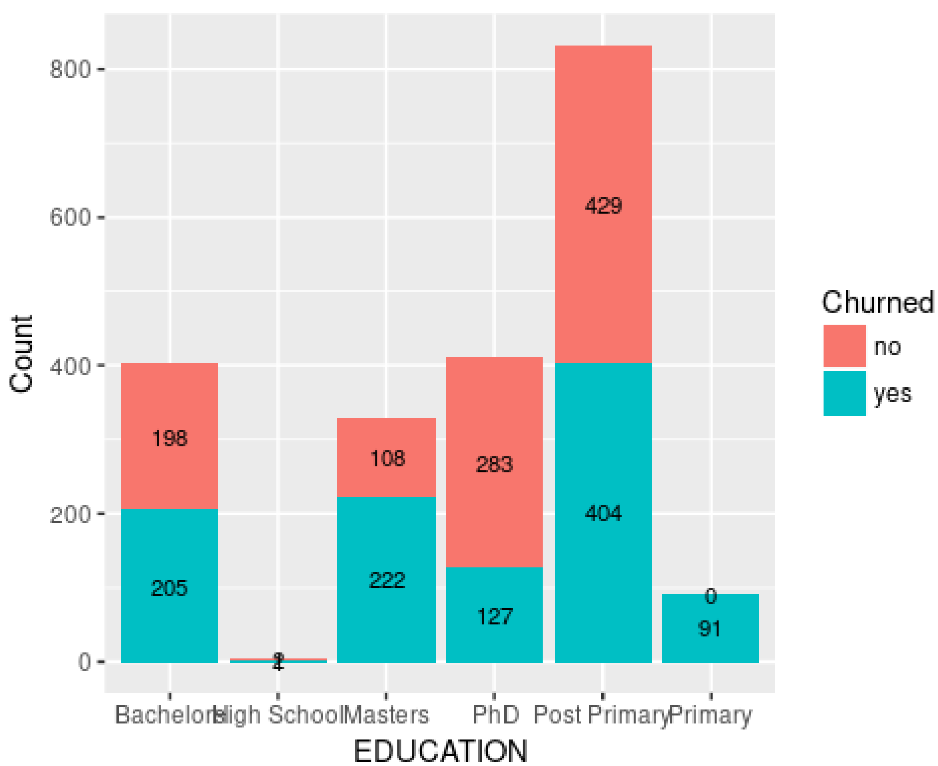
1.3.10 EDUCATION

Highest Level of education attainment the account holder has achieved

Predictor	EDUCATION
attribute type	**Nominal**
%Missing Values	**0.386%**
Mode	**Post Primary**

CHART CODE

```
plot_data <- data.frame(table(churn_data$EDUCATION, churn_data$CHURNER))
colnames(plot_data) <- c("EDUCATION", "Churned", "Count" )
ggplot(data=plot_data, aes(x = EDUCATION, y = Count, fill = Churned, label =
Count)) +
  geom_bar(stat = "identity") +
  geom_text(size = 3, position = position_stack(vjust = 0.5))
```



1.3.11 MINUTES_CURR_MONTH

Phone Minutes currently for current months (to the time the data was extracted)

Predictor	MINUTES_CURR_MONTH
attribute type	**Numeric**
%Missing Values	**0%**
Max	**14000**
Min	**1**
Mean	**748**
Mode	**2, 5**
median	**105**
Standard deviation	**2017.5556123**

MINUTES_CURR_MONTH Skewness

```
(3 * (mean(churn_data$MINUTES_PREV_MONTH) -
median(churn_data$MINUTES_PREV_MONTH)))/ sd(churn_data$MINUTES_PREV_MONTH)

## [1] 0.9312979
```

CHART CODE

```
plot_data <- data.frame(table(churn_data$MINUTES_CURR_MONTH, churn_data$CHURNER))
colnames(plot_data) <- c("MINUTES_CURR_MONTH", "CHURNER", "COUNT")
ggplot(data=plot_data, aes(x = MINUTES_CURR_MONTH, y=COUNT, fill = CHURNER,
label=COUNT)) +
  geom_histogram(stat="identity", width = 1) +
  geom_text(size = 2, position = position_stack(vjust = 0.5))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

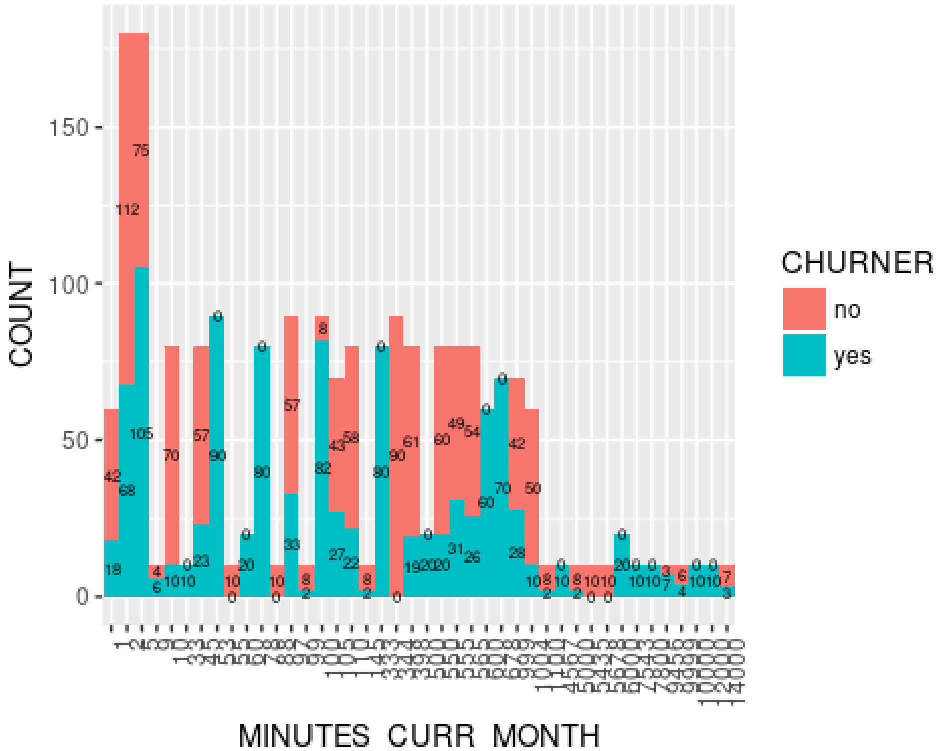
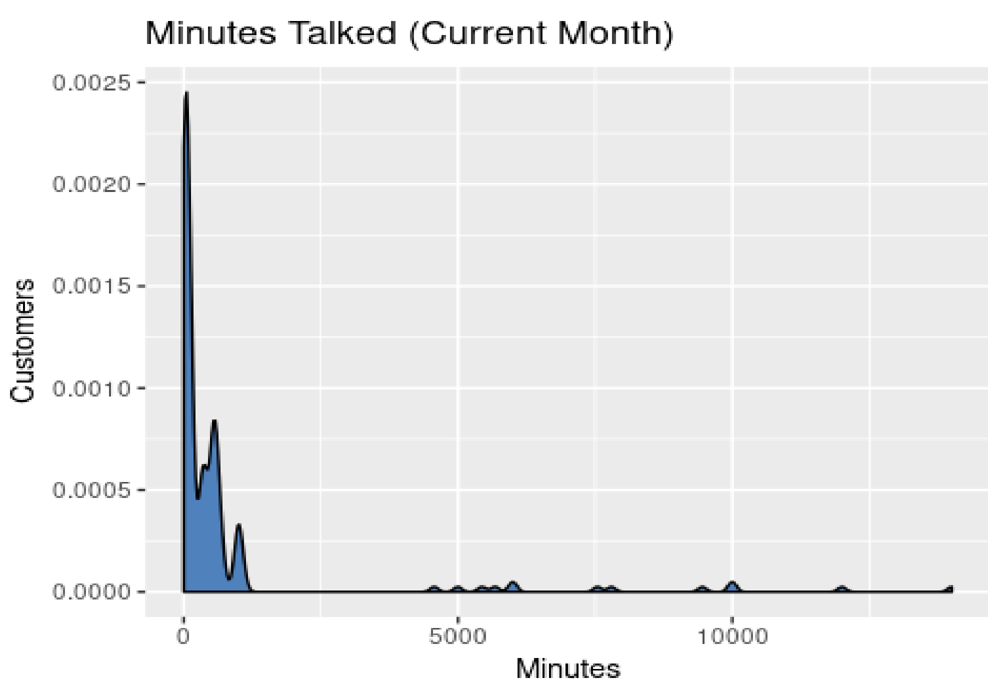


CHART CODE

```
ggplot(data=churn_data, aes(x=MINUTES_CURR_MONTH)) + geom_density(adjust=1,
fill="#4F81BD") + ggtitle("Minutes Talked (Current Month)") + xlab("Minutes") +
ylab("Customers")
```



1.3.12 MINUTES_PREV_MONTH

Phone Minutes used in the previous month

Predictor	MINUTES_PREV_MONTH
attribute type	Numerical
%Missing Values	0%
Max	16754
Min	0
Mean	864.31
Mode	0
median	98

MINUTES_PREV_MONTH Skewness

```
(3 * (mean(churn_data$MINUTES_PREV_MONTH) -
median(churn_data$MINUTES_PREV_MONTH)))/ sd(churn_data$MINUTES_PREV_MONTH)

## [1] 0.9312979
```

CHART CODE

```
plot_data <- data.frame(table(churn_data$MINUTES_PREV_MONTH, churn_data$CHURNER))
colnames(plot_data) <- c("MINUTES_PREV_MONTH", "CHURNER", "COUNT")
ggplot(data=plot_data, aes(x = MINUTES_PREV_MONTH, y=COUNT, fill = CHURNER,
label=COUNT)) +
  geom_histogram(stat="identity", width = 1) +
  geom_text(size = 2, position = position_stack(vjust = 0.5))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

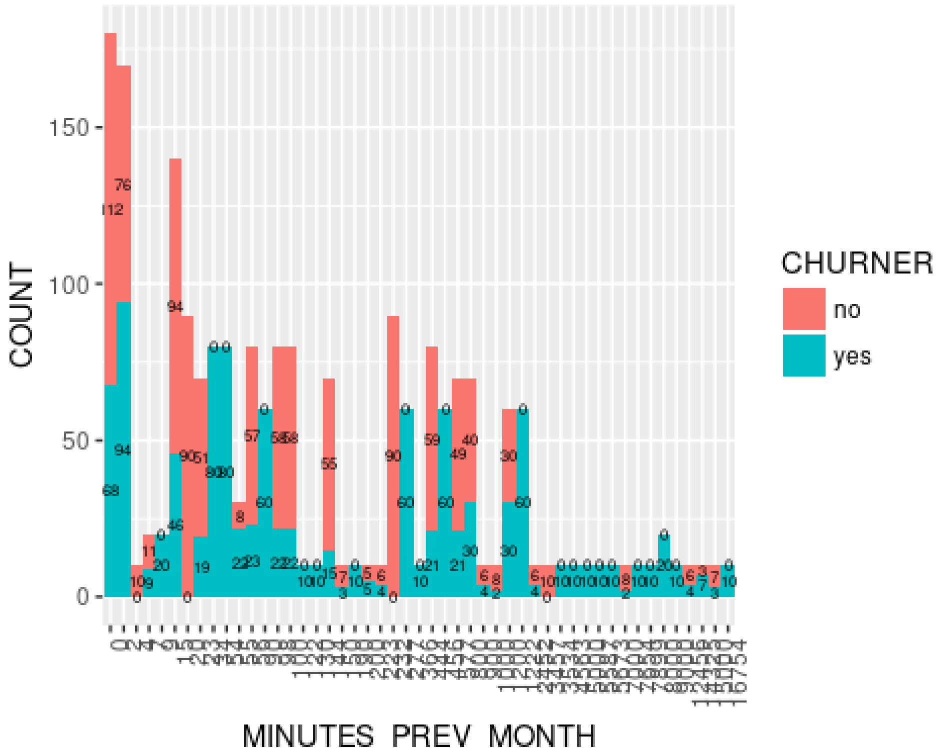
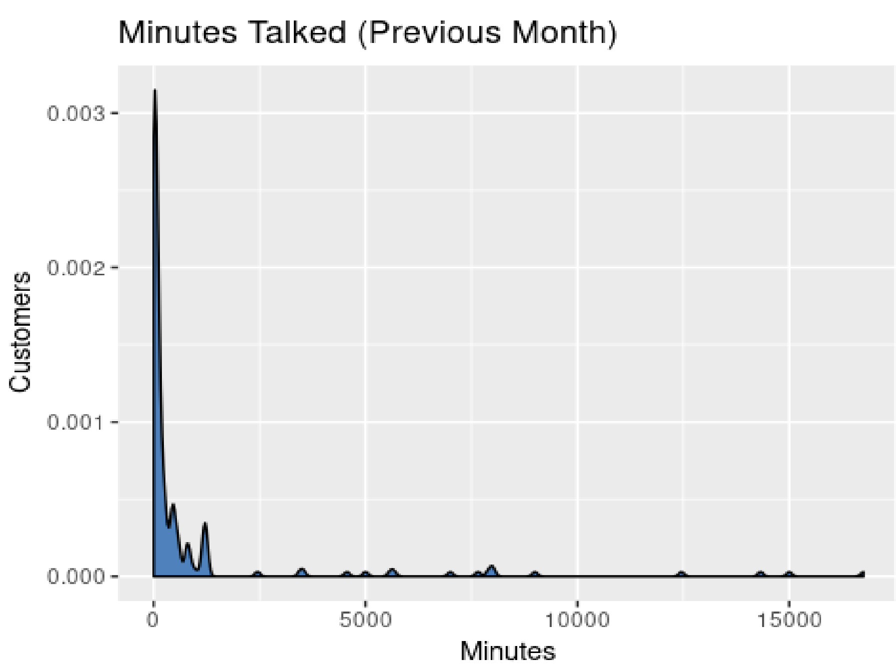


CHART CODE

```
ggplot(data=churn_data, aes(x=MINUTES_PREV_MONTH)) + geom_density(adjust=1,
fill="#4F81BD") + ggtitle("Minutes Talked (Previous Month)") + xlab("Minutes") +
ylab("Customers")
```



1.3.13 MINUTES_3MONTHS_AGO

Phone Minutes used in the 3 month PREVIOUS

Predictor	MINUTES_3MONTHS_AGO
attribute type	Numerical
%Missing Values	0.145%
Max	12456
Min	0
Mean	452.74
Mode	0
median	97

MINUTES_3MONTHS_AGO Skewness

Right-skewness data – Is positive, as mean is greater than the median

```
(3 * (mean(churn_data$MINUTES_3MONTHS_AGO) -
median(churn_data$MINUTES_3MONTHS_AGO)))/ sd(churn_data$MINUTES_3MONTHS_AGO)

## [1] 0.9012361
```

CHART CODE

```
plot_data <- data.frame(table(churn_data$MINUTES_3MONTHS_AGO,
churn_data$CHURNER))
colnames(plot_data) <- c("MINUTES_3MONTHS_AGO", "CHURNER", "COUNT")
ggplot(data=plot_data, aes(x = MINUTES_3MONTHS_AGO, y=COUNT, fill = CHURNER,
label=COUNT)) +
  geom_histogram(stat="identity", width = 1) +
  geom_text(size = 2, position = position_stack(vjust = 0.5))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

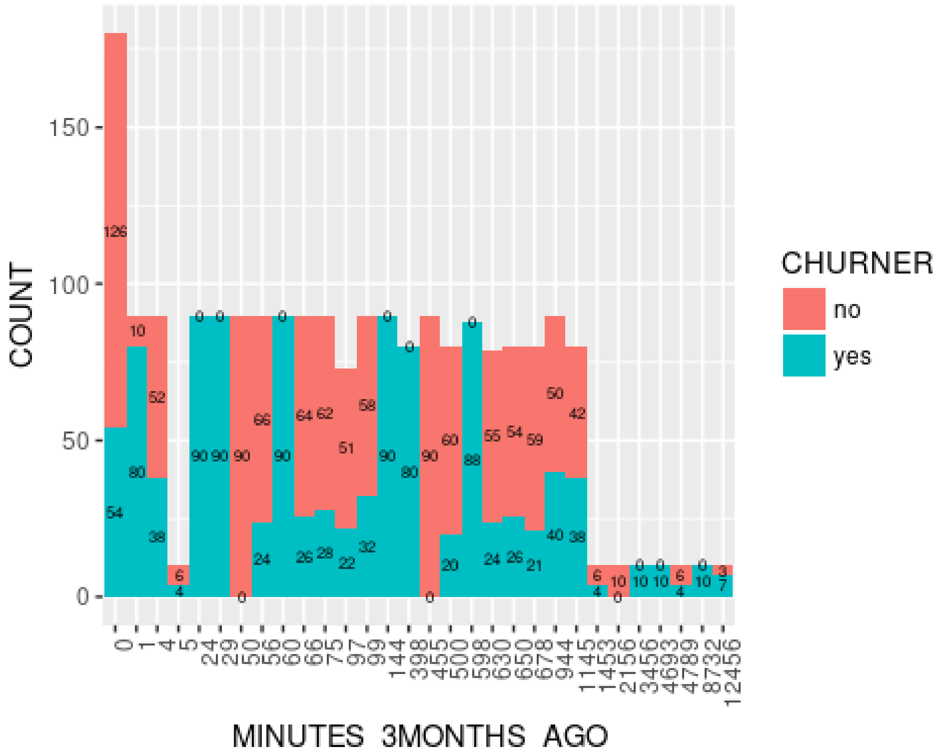
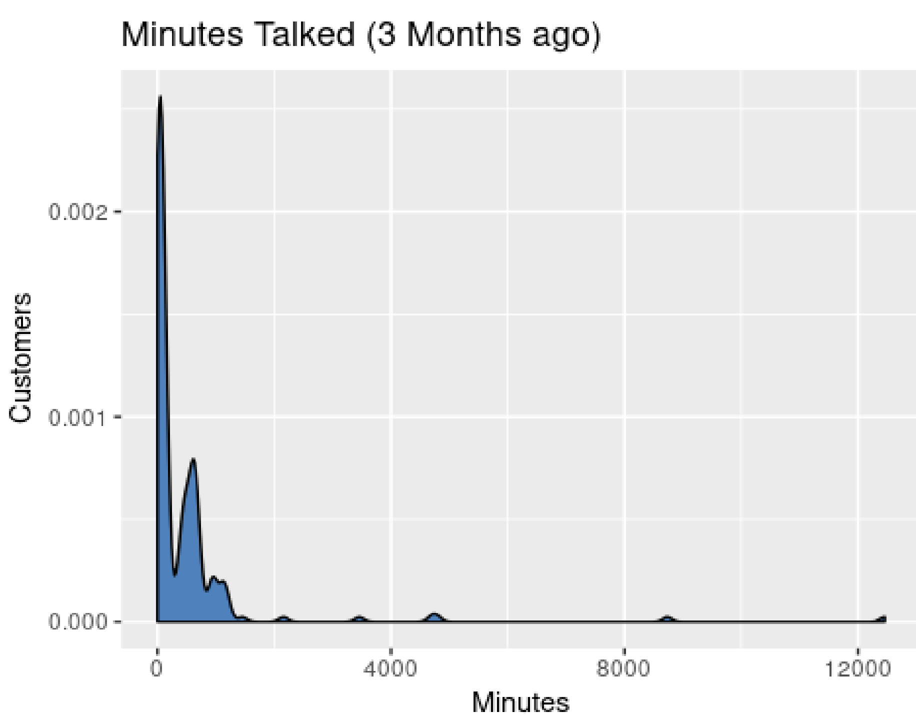


CHART CODE

```
ggplot(data=churn_data, aes(x=MINUTES_3MONTHS_AGO)) + geom_density(adjust=1,
fill="#4F81BD") + ggtitle("Minutes Talked (3 Months ago)") + xlab("Minutes") +
ylab("Customers")
```



1.3.14 CUST_MOS

The number of continuous months the Customer is with the provider

Predictor	CUST_MOS
attribute type	Numerical
%Missing Values	0.145%
Max	50
Min	1
Mean	16.05
Mode	11
Mode	11
median	11

CUST_MOS Skewness

Right-skewness data – Is positive, as mean is greater than the median

```
(3 * (mean(churn_data$CUST_MOS) - median(churn_data$CUST_MOS)))/
sd(churn_data$CUST_MOS)

## [1] 1.132926
```

CHART CODE

```
plot_data <- data.frame(table(churn_data$CUST_MOS, churn_data$CHURNER))
colnames(plot_data) <- c("CUST_MOS", "CHURNER", "COUNT")
ggplot(data=plot_data, aes(x = CUST_MOS, y=COUNT, fill = CHURNER, label=COUNT)) +
  geom_histogram(stat="identity", width = 1) +
  geom_text(size = 2, angle = 90, position = position_stack(vjust = 0.5))+
  theme(axis.text.x = element_text(size=6, hjust = .5))

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

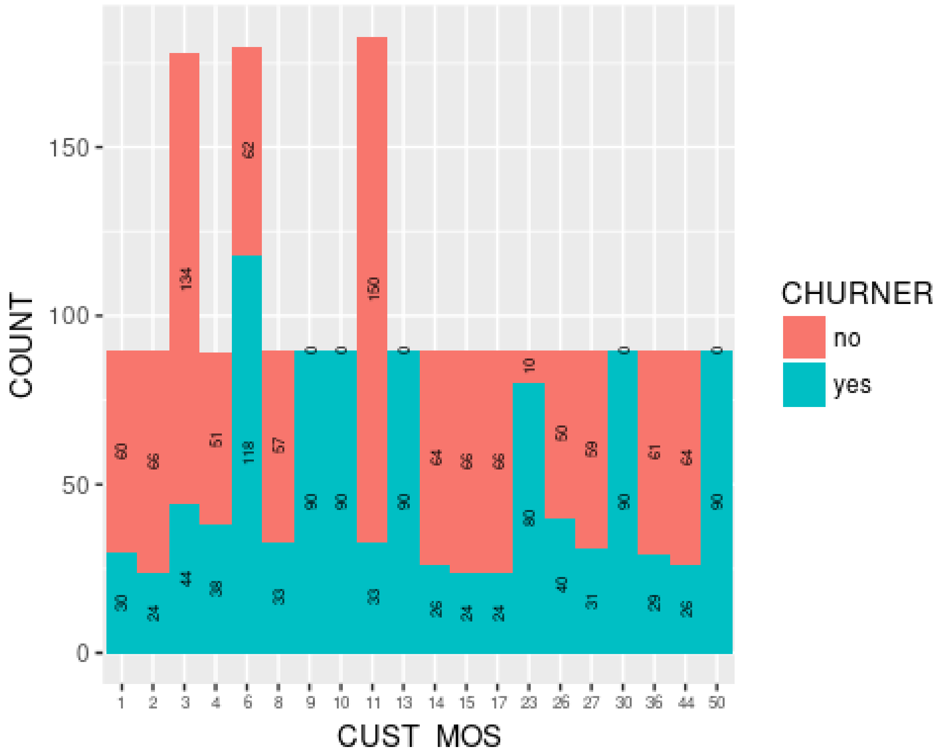
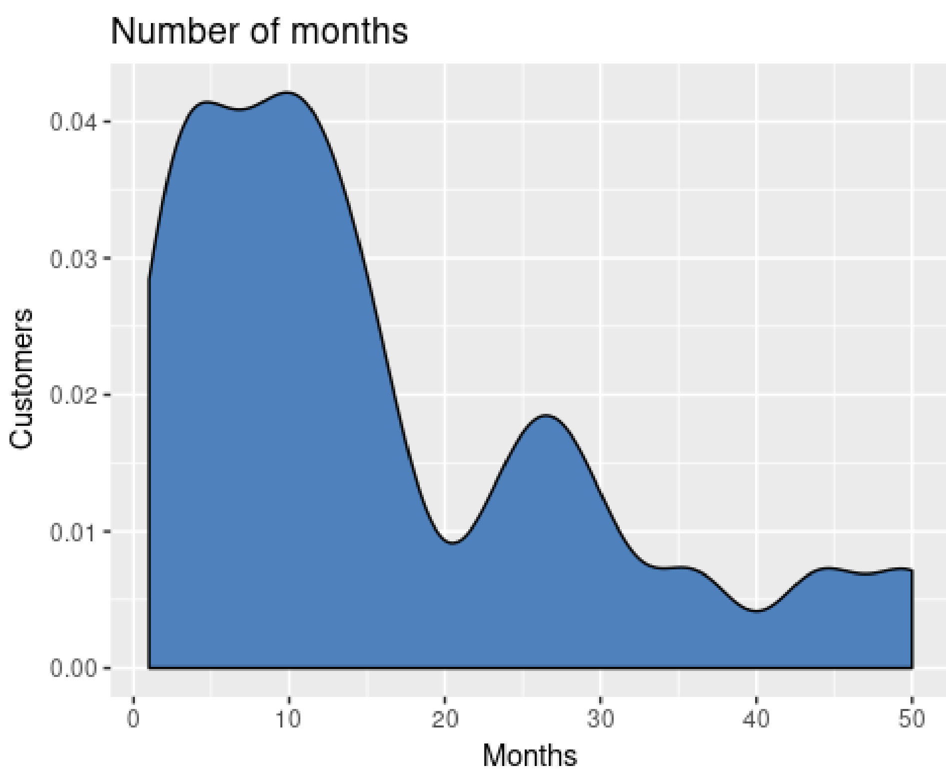


CHART CODE

```
ggplot(data=churn_data, aes(x=CUST_MOS)) + geom_density(adjust=1, fill="#4F81BD") +
ggtitle("Number of months") + xlab("Months") + ylab("Customers")
```



1.3.15 TOT_MINUTES_USAGE

The total number of minutes used to date

Predictor	TOT_MINUTES_USAGE
attribute type	Numeric
%Missing Values	0.193%
Max	36237
Min	0
Mean	2037.09
Mode	0
median	264
Standard deviation	4883.9078032

TOT_MINUTES_USAGE Skewness

Right-skewness data – Is positive, as mean is greater than the median

```
(3 * (mean(churn_data$TOT_MINUTES_USAGE) -
median(churn_data$TOT_MINUTES_USAGE)))/ sd(churn_data$TOT_MINUTES_USAGE)

## [1] 1.089144
```

CHART CODE

```
plot_data <- data.frame(table(churn_data$TOT_MINUTES_USAGE, churn_data$CHURNER))
colnames(plot_data) <- c("TOT_MINUTES_USAGE", "CHURNER", "COUNT")
ggplot(data=plot_data, aes(x = TOT_MINUTES_USAGE, y=COUNT, fill = CHURNER,
label=COUNT)) +
  geom_histogram(stat="identity", width = 1) +
  geom_text(size = 2, angle = 90, position = position_stack(vjust = 0.5))+
  theme(axis.text.x = element_text(size=6, angle = 90, hjust = .5))

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

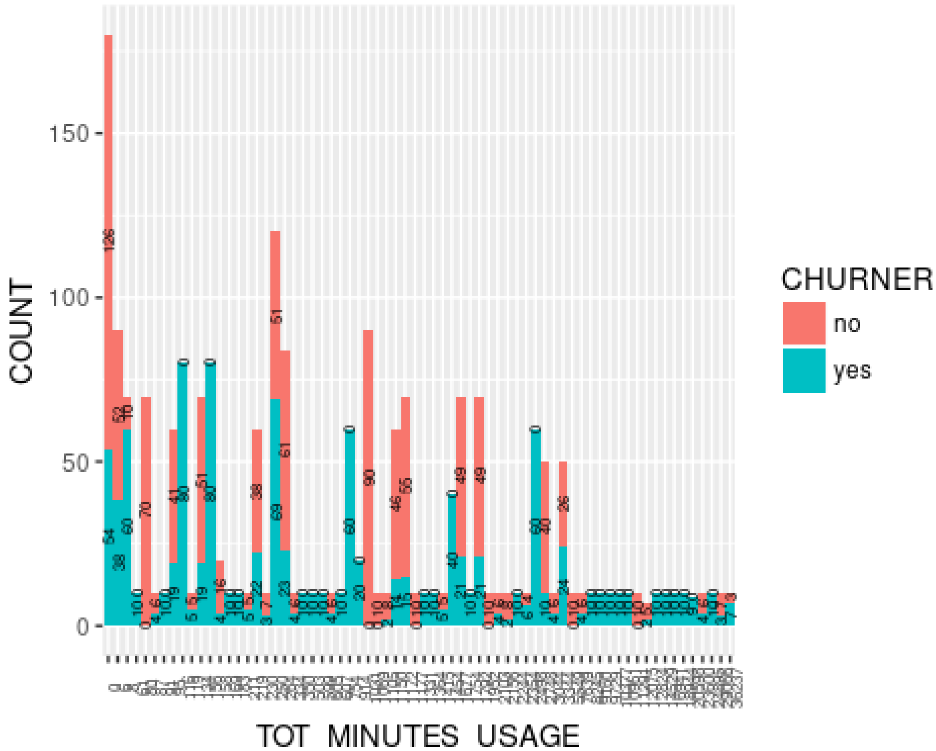
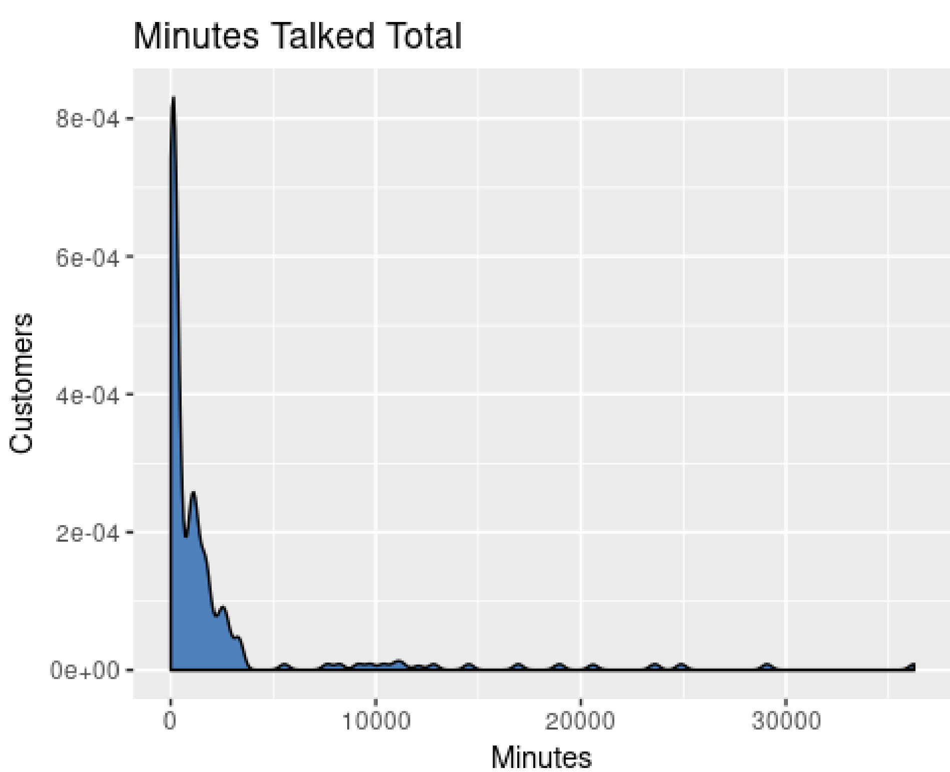


CHART CODE

```
ggplot(data=churn_data, aes(x=TOT_MINUTES_USAGE)) + geom_density(adjust=1,
fill="#4F81BD") + ggtitle("Minutes Talked Total") + xlab("Minutes") + ylab("Customers")
```



1.3.16 NUM_LINES

The number of fixed lines the customer has leased.

Predictor	NUM_LINES
attribute type	Numeric
%Missing Values	0%
Max	3
Min	1
Mean	1.3913043
Mode	1
median	1
Standard deviation	0.5703498

NUM_LINES Skewness

Right-skewness data – Is positive, as mean is greater than the median

```
(3 * (mean(churn_data$NUM_LINES) - median(churn_data$NUM_LINES)))/
sd(churn_data$NUM_LINES)

## [1] 2.058233
```

CHART CODE

```
plot_data <- data.frame(table(churn_data$NUM_LINES, churn_data$CHURNER))
colnames(plot_data) <- c("NUM_LINES", "CHURNER", "COUNT")
ggplot(data=plot_data, aes(x = NUM_LINES, y=COUNT, fill = CHURNER, label=COUNT))
+
  geom_histogram(stat="identity", width = 1) +
  geom_text(size = 2, position = position_stack(vjust = 0.5))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

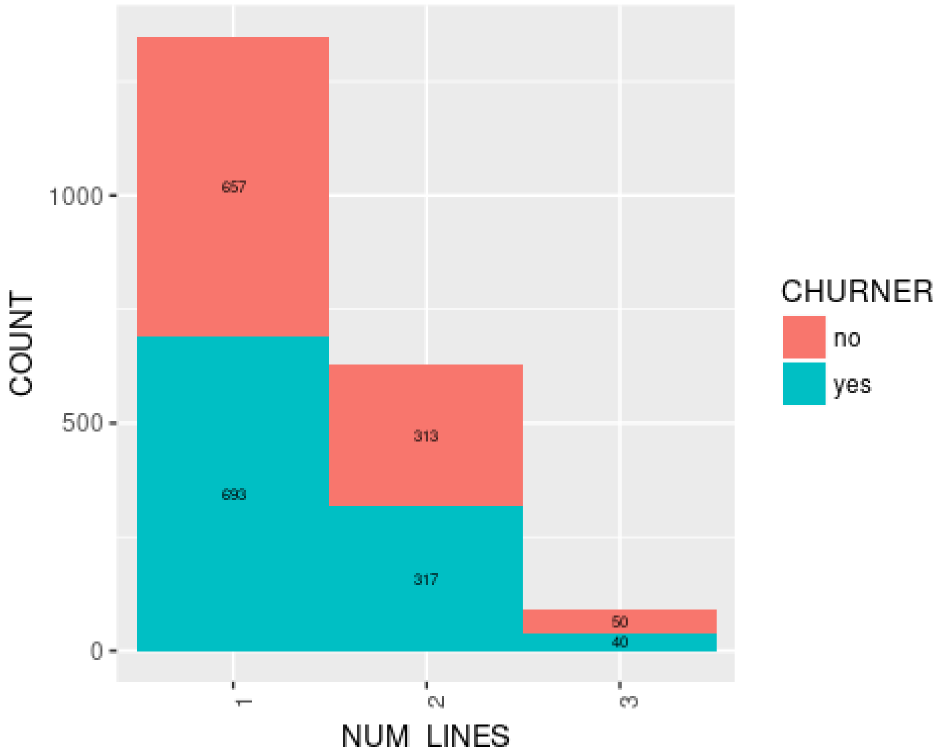
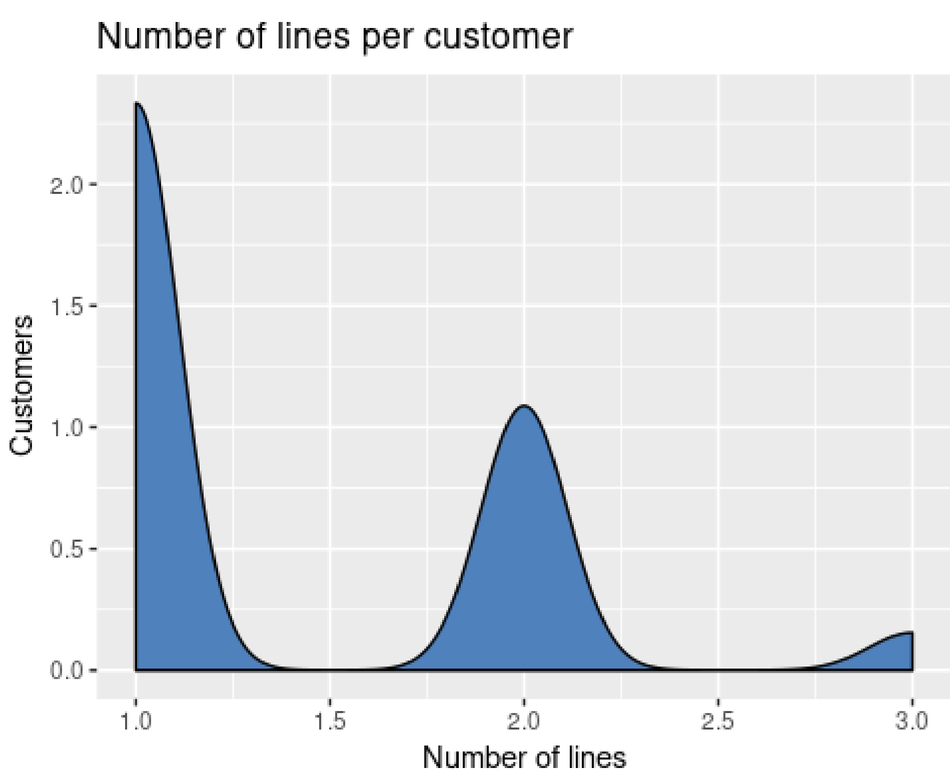


CHART CODE

```
ggplot(data=churn_data, aes(x=NUM_LINES)) + geom_density(adjust=1, fill="#4F81BD") +
ggtitle("Number of lines per customer") + xlab("Number of lines") + ylab("Customers")
```

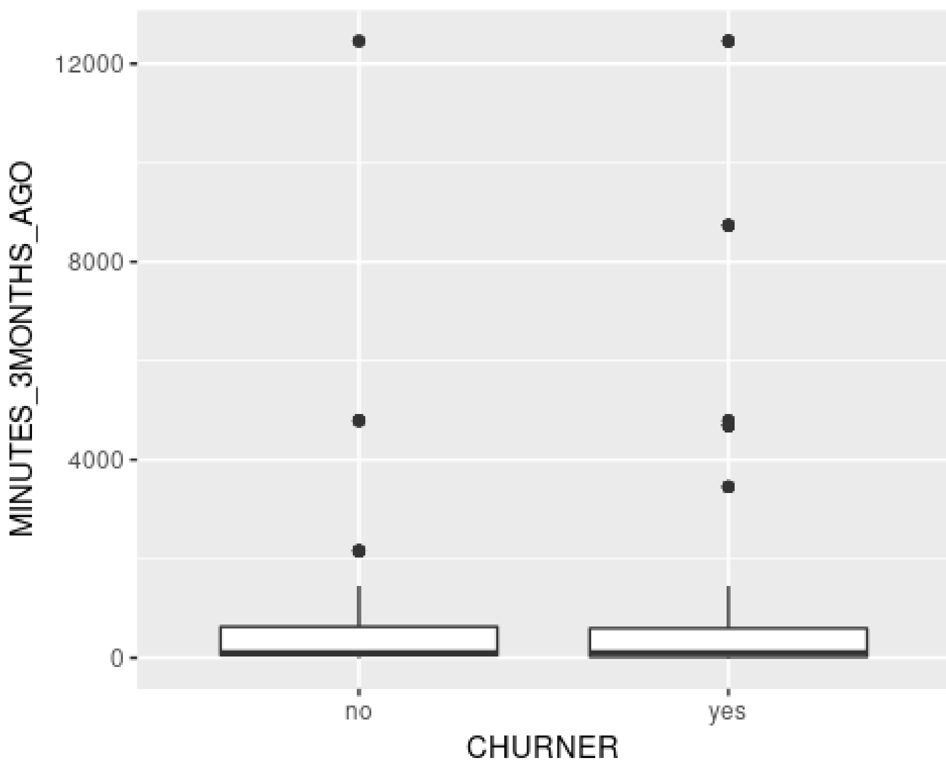


1.4 Identify Outliers

1.4.1 Graphical methods to identify outliers

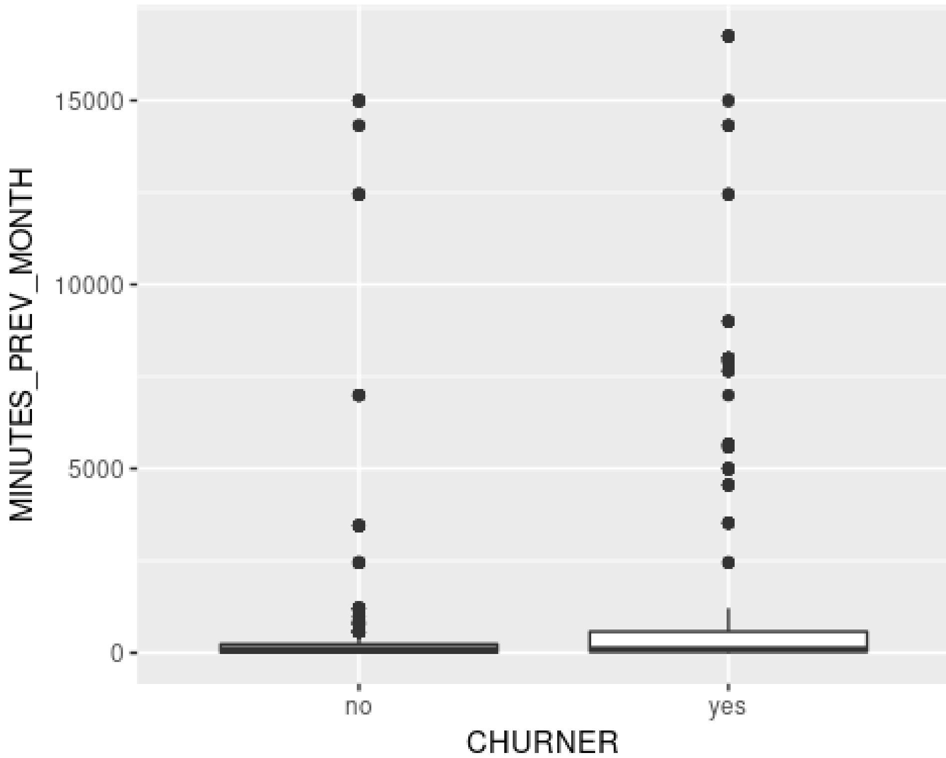
1.4.1.1 MINUTES_3MONTHS_AGO

```
ggplot(data=churn_data, aes(x=CHURNER, y=MINUTES_3MONTHS_AGO))+ geom_boxplot()
```



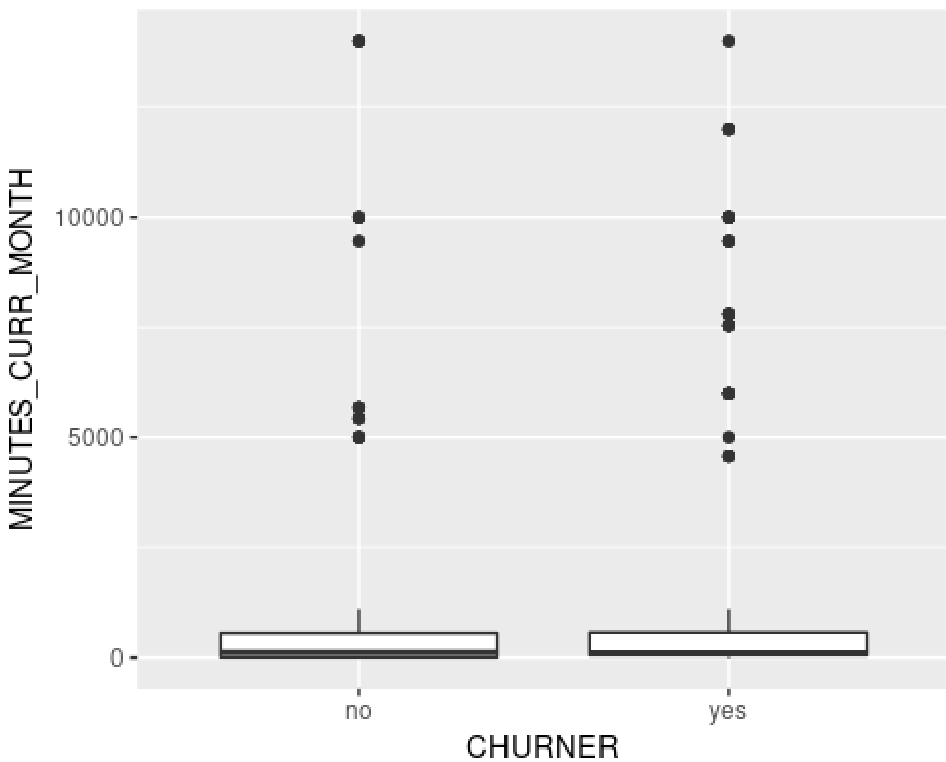
1.4.1.2 MINUTES_PREV_MONTH

```
ggplot(data=churn_data, aes(x=CHURNER, y=MINUTES_PREV_MONTH))+ geom_boxplot()
```



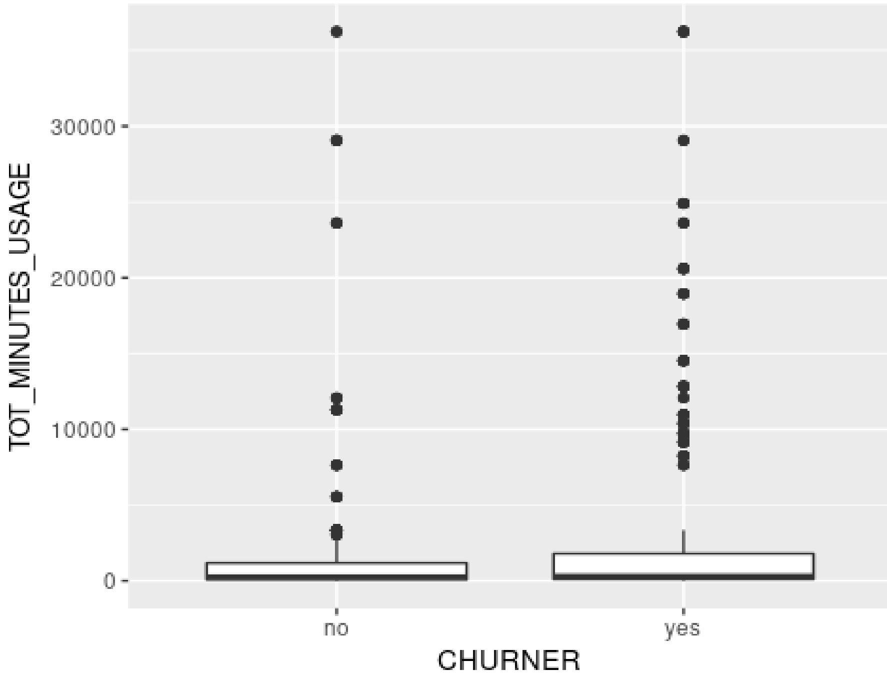
1.4.1.3 MINUTES_CURR_MONTH

```
ggplot(data=churn_data, aes(x=CHURNER, y=MINUTES_CURR_MONTH))+ geom_boxplot()
```



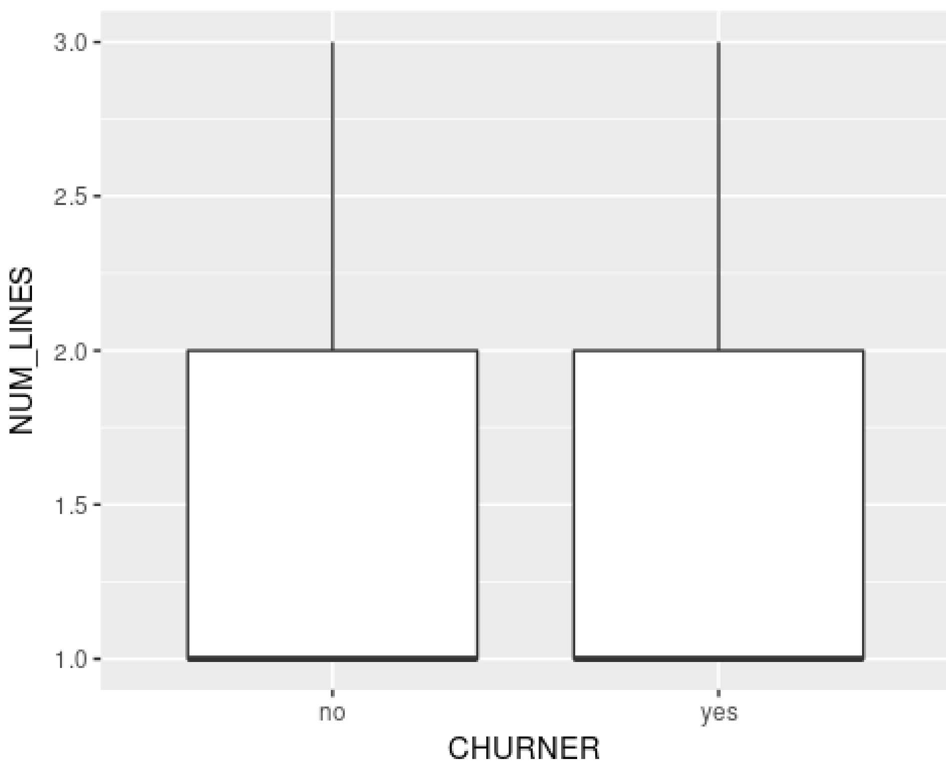
1.4.1.4 TOT_MINUTES_USAGE

```
ggplot(data=churn_data, aes(x=CHURNER, y=TOT_MINUTES_USAGE))+ geom_boxplot()
```



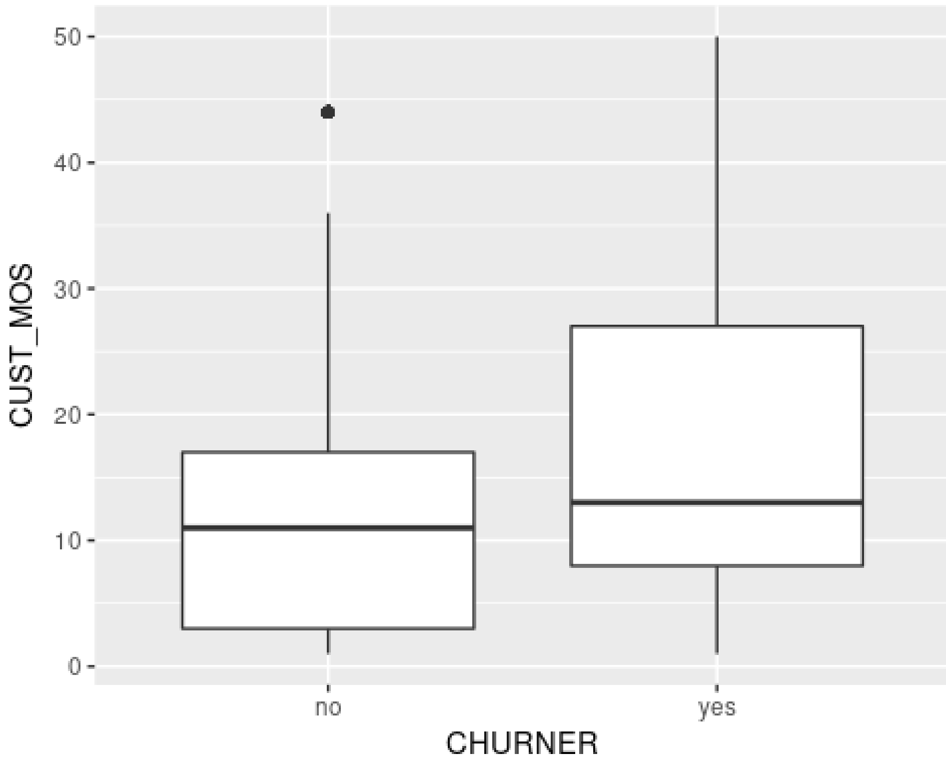
1.4.1.5 NUM_LINES

```
ggplot(data=churn_data, aes(x=CHURNER, y=NUM_LINES))+ geom_boxplot()
```



1.4.1.6 CUST_MOS

```
ggplot(data=churn_data, aes(x=CHURNER, y=CUST_MOS))+ geom_boxplot()
```



1.4.2 Mathematically identifying outliers

Choose a numeric predictor variable that has possible outliers based on your analysis in 2 above. Use the IQR method and Z-Score Standardisation method to identify outliers. Discuss your findings.

IQR method

```
minutesQuartileTolerance <- 1.5* IQR(churn_data$TOT_MINUTES_USAGE)
minutesLowerQuantile <- quantile(churn_data$TOT_MINUTES_USAGE, 0.25)
minutesUpperQuantile <- quantile(churn_data$TOT_MINUTES_USAGE, 0.75)

print("lower outliers")

## [1] "lower outliers"

churn_data$TOT_MINUTES_USAGE[churn_data$TOT_MINUTES_USAGE < (minutesLowerQuantile
- minutesQuartileTolerance)]

## numeric(0)

print("upper outliers")

## [1] "upper outliers"
```

```
churn_data$TOT_MINUTES_USAGE[churn_data$TOT_MINUTES_USAGE > (minutesUpperQuantile
+ minutesQuartileTolerance)]

##   [1] 18944 14529 12824 23600 10961 20598 29066  9160 12075 11291  7639
##  [12] 36237  9729 16941 10377 11291  7639 36237  9729  5549 16941 10377
##  [23]  5549 24895  8245  5549 24895  8245 16941 10377  5549 24895  8245
##  [34] 18944 14529 12824 23600 10961 20598 16941 10377 24895  8245  5549
##  [45] 18944 14529 12824 23600 10961 20598 29066  9160  5549 12075 11291
##  [56] 18944 14529 12824 23600 10961 20598 29066  9160 11291  7639 36237
##  [67]  9729 29066  9160 12075 11291  7639 36237  9729 24895  8245  7639
##  [78] 36237  9729 16941 10377 24895  8245  5549 18944 14529 12824 23600
##  [89] 10961 29066  5549  9160 12075 11291  7639 36237  9729  5549 16941
## [100] 10377 18944 14529 12824  5549 23600 10961 20598 29066  9160 11291
## [111]  7639 36237  9729 16941 29066  9160 12075 11291  7639 36237  9729
## [122] 16941 10377 24895 10961 20598 29066  9160 11291  7639 36237  9729
## [133] 16941 10377 24895  8245 10377 24895  8245 18944 14529 12824 23600
## [144] 10961 20598  8245 18944 14529 12824 23600 24895  8245 18944 14529
## [155] 12824 23600 10961 20598 29066  9160 12075 18944 14529 12824 23600
## [166] 10961 20598 29066  9160 12075 11291  7639 36237  9729 16941 10377
```

zScore Standardisation method

```
# zScore Standardisation
zscore.TOT_MINUTES_USAGE <-(churn_data$TOT_MINUTES_USAGE -
mean(churn_data$TOT_MINUTES_USAGE))/sd(churn_data$TOT_MINUTES_USAGE)
summary(zscore.TOT_MINUTES_USAGE)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.41710 -0.39340 -0.36300  0.00000 -0.07373  7.00300

zscore.TOT_MINUTES_USAGE[zscore.TOT_MINUTES_USAGE < -3 | zscore.TOT_MINUTES_USAGE
> 3]

##  [1] 3.461758 4.415093 3.800421 5.534279 7.002570 3.051636 7.002570
##  [8] 3.051636 4.680249 4.680249 3.051636 4.680249 3.461758 4.415093
## [15] 3.800421 3.051636 4.680249 3.461758 4.415093 3.800421 5.534279
## [22] 3.461758 4.415093 3.800421 5.534279 7.002570 5.534279 7.002570
## [29] 4.680249 7.002570 3.051636 4.680249 3.461758 4.415093 5.534279
## [36] 7.002570 3.051636 3.461758 4.415093 3.800421 5.534279 7.002570
## [43] 3.051636 5.534279 7.002570 3.051636 4.680249 3.800421 5.534279
## [50] 7.002570 3.051636 4.680249 4.680249 3.461758 4.415093 3.800421
## [57] 3.461758 4.415093 4.680249 3.461758 4.415093 3.800421 5.534279
## [64] 3.461758 4.415093 3.800421 5.534279 7.002570 3.051636
```

1.5 Skewness for numeric data

Right-skewness data – Is positive, as mean is greater than the median Left skewness data – Mean is smaller than the median, generating negative values Perfectly symmetric data – mean, median and mode are equal, so skewness is zero

Skewness for `MINUTES_3MONTHS_AGO`

```
(3 * (mean(churn_data$MINUTES_3MONTHS_AGO) -
median(churn_data$MINUTES_3MONTHS_AGO)))/ sd(churn_data$MINUTES_3MONTHS_AGO)

## [1] 0.9012361
```

Skewness for `MINUTES_PREV_MONTH`

```
(3 * (mean(churn_data$MINUTES_PREV_MONTH) -
median(churn_data$MINUTES_PREV_MONTH)))/ sd(churn_data$MINUTES_PREV_MONTH)

## [1] 0.9312979
```

Skewness for `MINUTES_CURR_MONTH`

```
(3 * (mean(churn_data$MINUTES_CURR_MONTH) -
median(churn_data$MINUTES_CURR_MONTH)))/ sd(churn_data$MINUTES_CURR_MONTH)

## [1] 0.956244
```

Skewness for `TOT_MINUTES_USAGE`

```
(3 * (mean(churn_data$TOT_MINUTES_USAGE) -
median(churn_data$TOT_MINUTES_USAGE)))/ sd(churn_data$TOT_MINUTES_USAGE)

## [1] 1.089144
```

Skewness for `NUM_LINES`

```
(3 * (mean(churn_data$NUM_LINES) - median(churn_data$NUM_LINES)))/
sd(churn_data$NUM_LINES)

## [1] 2.058233
```

Skewness for `CUST_MOS`

```
(3 * (mean(churn_data$CUST_MOS) - median(churn_data$CUST_MOS)))/
sd(churn_data$CUST_MOS)

## [1] 1.132926
```

1.5.1 Correcting skewness

```
# calculate skewness
TOT_MINUTES_USAGESkewness <- (3 * (mean(churn_data$TOT_MINUTES_USAGE) -
median(churn_data$TOT_MINUTES_USAGE)))/ sd(churn_data$TOT_MINUTES_USAGE)
TOT_MINUTES_USAGESkewness

## [1] 1.089144
```

Natural Log Transformation

```
# Natural Log Transformation
natlog.TOT_MINUTES_USAGE <-
log(churn_data$TOT_MINUTES_USAGE[churn_data$TOT_MINUTES_USAGE > 0])

# applying log function to skewness
logTOT_MINUTES_USAGESkewness <- (3 * (mean(natlog.TOT_MINUTES_USAGE) -
median(natlog.TOT_MINUTES_USAGE)))/ sd(natlog.TOT_MINUTES_USAGE)
logTOT_MINUTES_USAGESkewness

## [1] -0.7042918
```

Square Root Transformation

```
# Square Root Transformation
sqrt.TOT_MINUTES_USAGE <- sqrt(churn_data$TOT_MINUTES_USAGE)
# applying sqr root function to skewness

sqrTOT_MINUTES_USAGESkewness <- (3 * (mean(sqrt.TOT_MINUTES_USAGE) -
median(sqrt.TOT_MINUTES_USAGE)))/ sd(sqrt.TOT_MINUTES_USAGE)
sqrTOT_MINUTES_USAGESkewness

## [1] 1.289714
```

zScore Standardisation

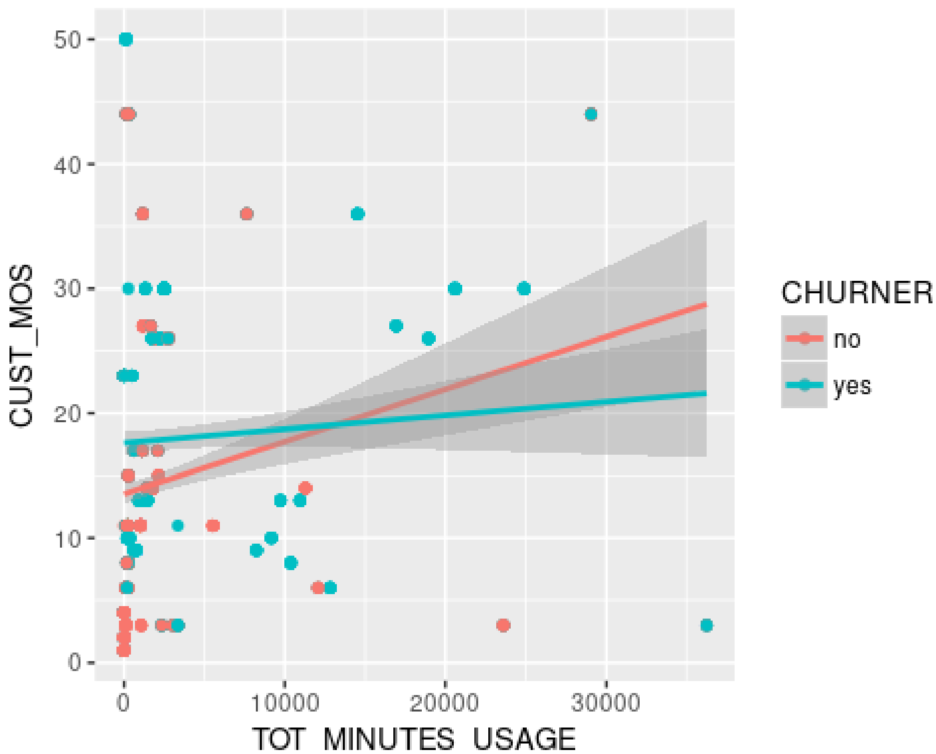
```
# zScore Standardisation
zscore.TOT_MINUTES_USAGE <-(churn_data$TOT_MINUTES_USAGE -
mean(churn_data$TOT_MINUTES_USAGE))/sd(churn_data$TOT_MINUTES_USAGE)

# applying zScore function to skewness
zscore.TOT_MINUTES_USAGE <- (3 * (mean(zscore.TOT_MINUTES_USAGE) -
median(zscore.TOT_MINUTES_USAGE)))/ sd(zscore.TOT_MINUTES_USAGE)
zscore.TOT_MINUTES_USAGE

## [1] 1.089144
```

1.7 2D Scatter plots to investigate correlation

2D Scatter plots to investigate correlation between numeric variables

1.7.1 TOT_MINUTES_USAGE vs CUST_MOS

```
ggplot(churn_data, aes(x=TOT_MINUTES_USAGE, y=CUST_MOS, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

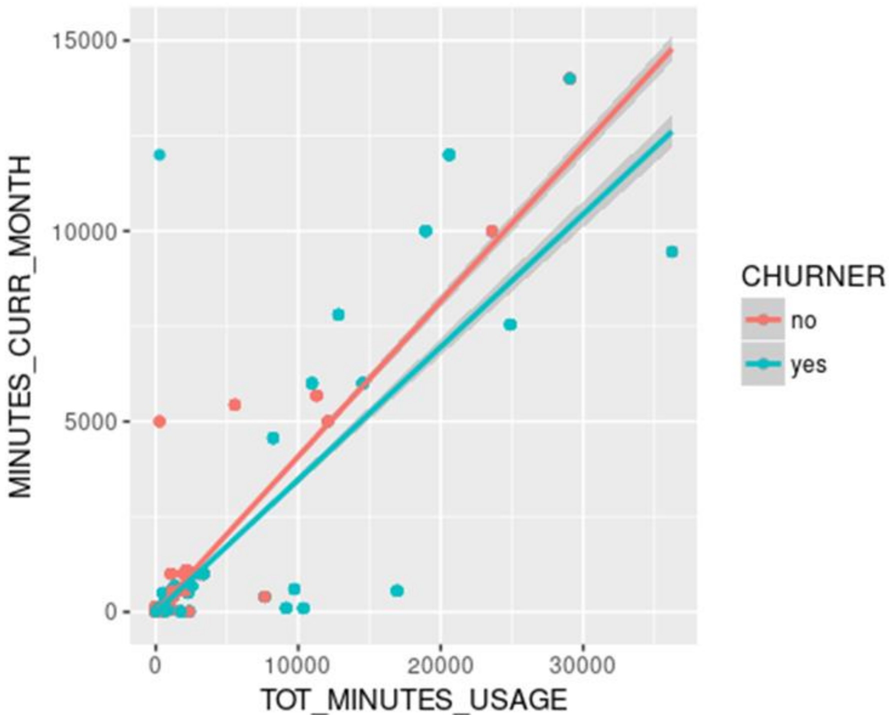


Correlation

```
cor(x=churn_data$TOT_MINUTES_USAGE, y=churn_data$CUST_MOS, use="all.obs",
method="pearson")

## [1] 0.09055857

# http://www.statmethods.net/stats/correlations.html
```

1.7.2 TOT_MINUTES_USAGE vs MINUTES_CURR_MONTH

```
ggplot(churn_data, aes(x=TOT_MINUTES_USAGE, y=MINUTES_CURR_MONTH, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

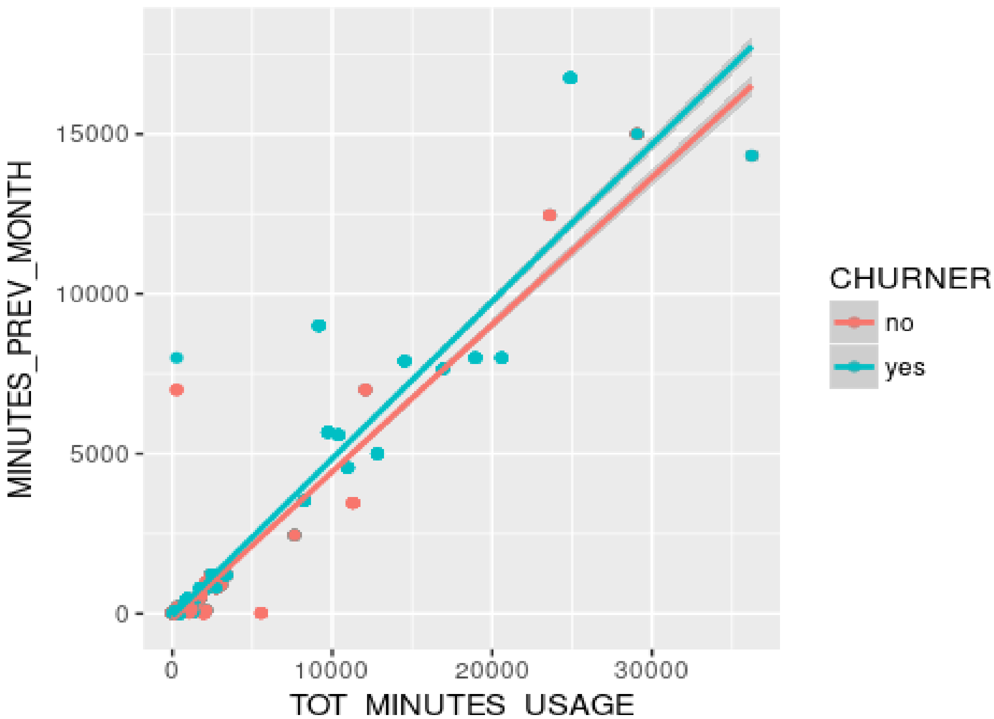


Correlation

```
cor(x=churn_data$TOT_MINUTES_USAGE, y=churn_data$MINUTES_CURR_MONTH,
use="all.obs", method="pearson")

## [1] 0.8844237

# http://www.statmethods.net/stats/correlations.html
```

1.7.3 TOT_MINUTES_USAGE vs MINUTES_PREV_MONTH

```
ggplot(churn_data, aes(x=TOT_MINUTES_USAGE, y=MINUTES_PREV_MONTH, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

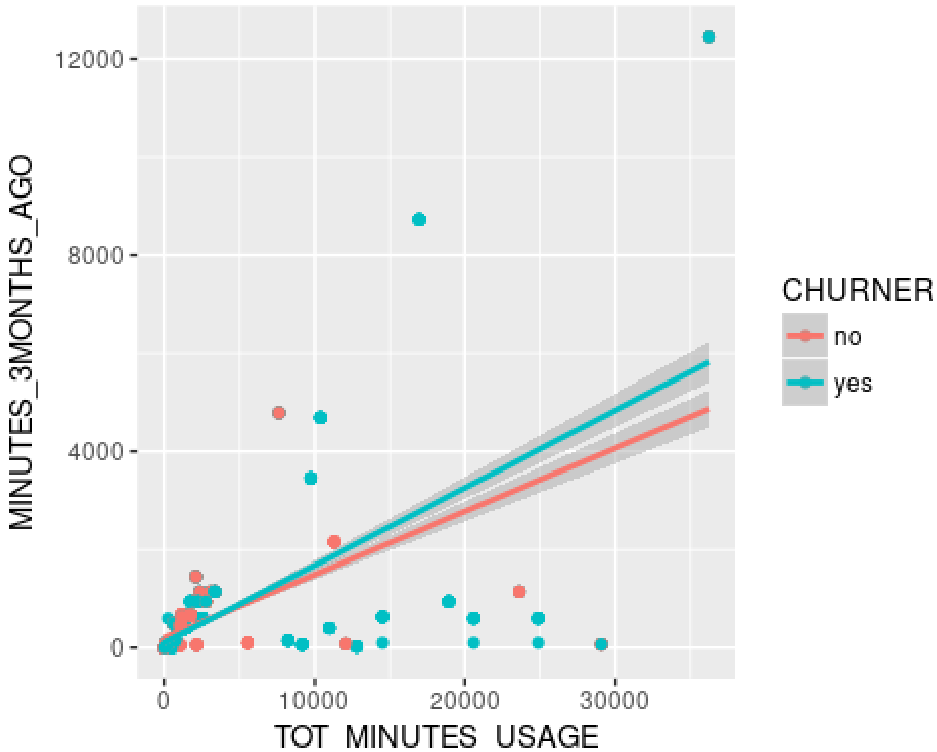


Correlation

```
cor(x=churn_data$TOT_MINUTES_USAGE, y=churn_data$MINUTES_PREV_MONTH,
use="all.obs", method="pearson")

## [1] 0.9568583

# http://www.statmethods.net/stats/correlations.html
```

1.7.4 TOT_MINUTES_USAGE vs MINUTES_3MONTHS_AGO

```
ggplot(churn_data, aes(x=TOT_MINUTES_USAGE, y=MINUTES_3MONTHS_AGO, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

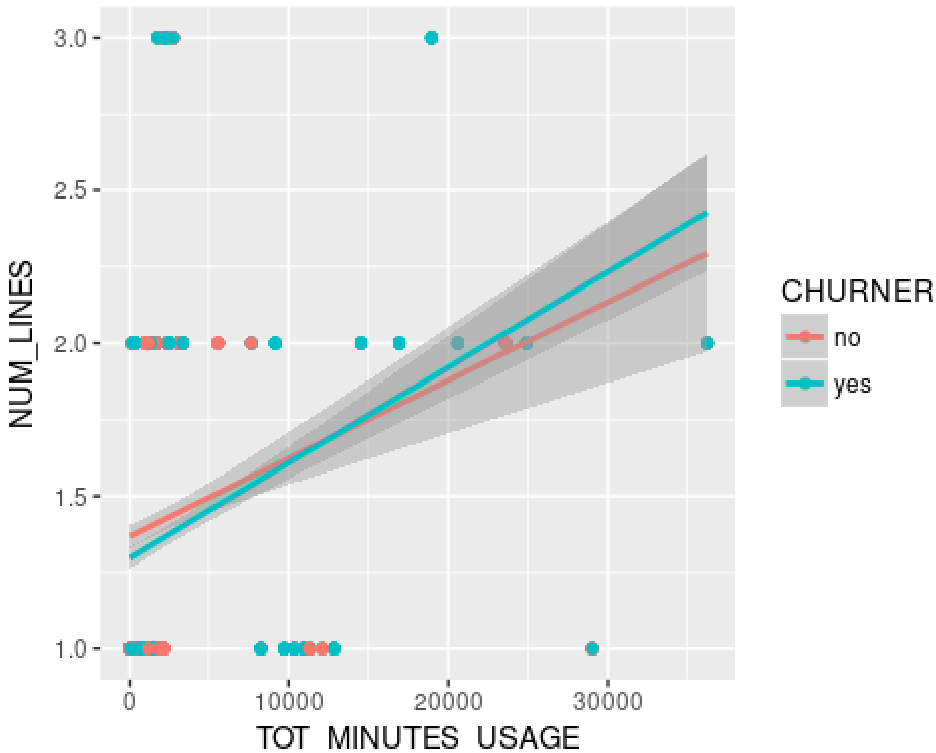


Correlation

```
cor(x=churn_data$TOT_MINUTES_USAGE, y=churn_data$MINUTES_3MONTHS_AGO,
use="all.obs", method="pearson")

## [1] 0.611558

# http://www.statmethods.net/stats/correlations.html
```

1.7.5 TOT_MINUTES_USAGE vs NUM_LINES

```
ggplot(churn_data, aes(x=TOT_MINUTES_USAGE, y=NUM_LINES, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

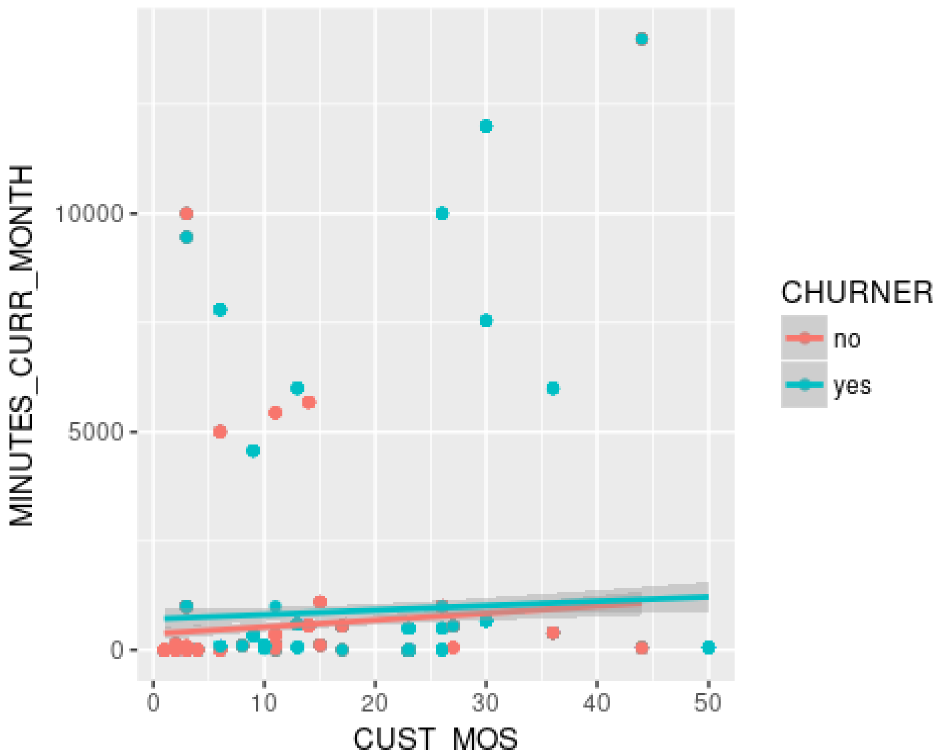


Correlation

```
cor(x=churn_data$TOT_MINUTES_USAGE, y=churn_data$NUM_LINES, use="all.obs",
method="pearson")

## [1] 0.2460581

# http://www.statmethods.net/stats/correlations.html
```

1.7.6 CUST_MOS vs MINUTES_CURR_MONTH

```
ggplot(churn_data, aes(x=CUST_MOS, y=MINUTES_CURR_MONTH, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

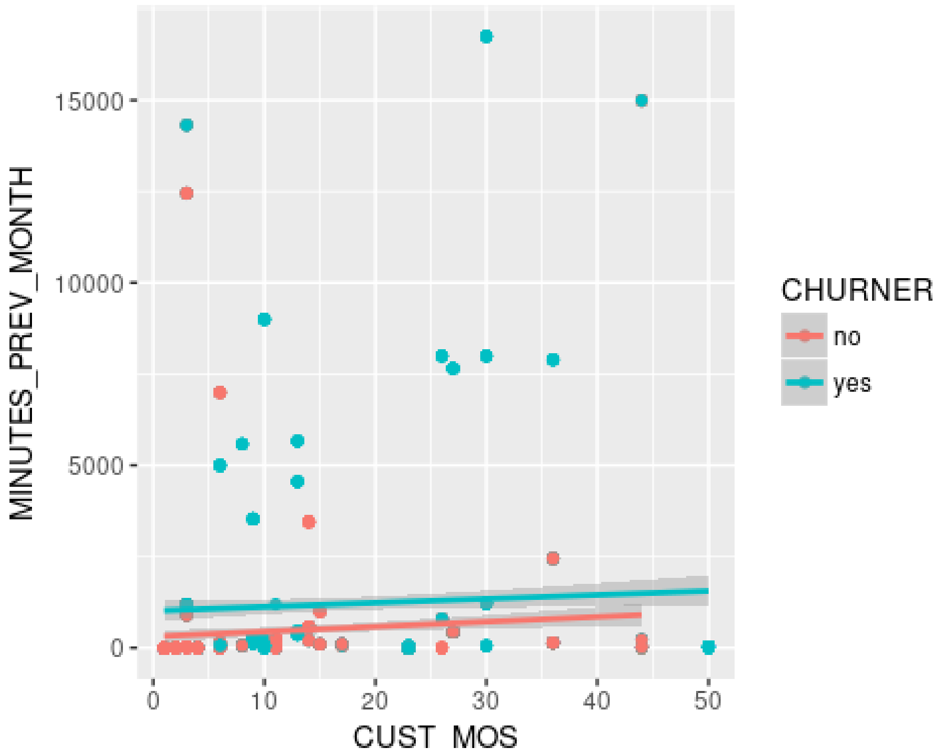


Correlation

```
cor(x=churn_data$TOT_MINUTES_USAGE, y=churn_data$MINUTES_CURR_MONTH,
use="all.obs", method="pearson")

## [1] 0.8844237

# http://www.statmethods.net/stats/correlations.html
```

1.7.7 CUST_MOS vs MINUTES_PREV_MONTH

```
ggplot(churn_data, aes(x=CUST_MOS, y=MINUTES_PREV_MONTH)) +
    geom_point() +
    geom_smooth(method=lm)
```

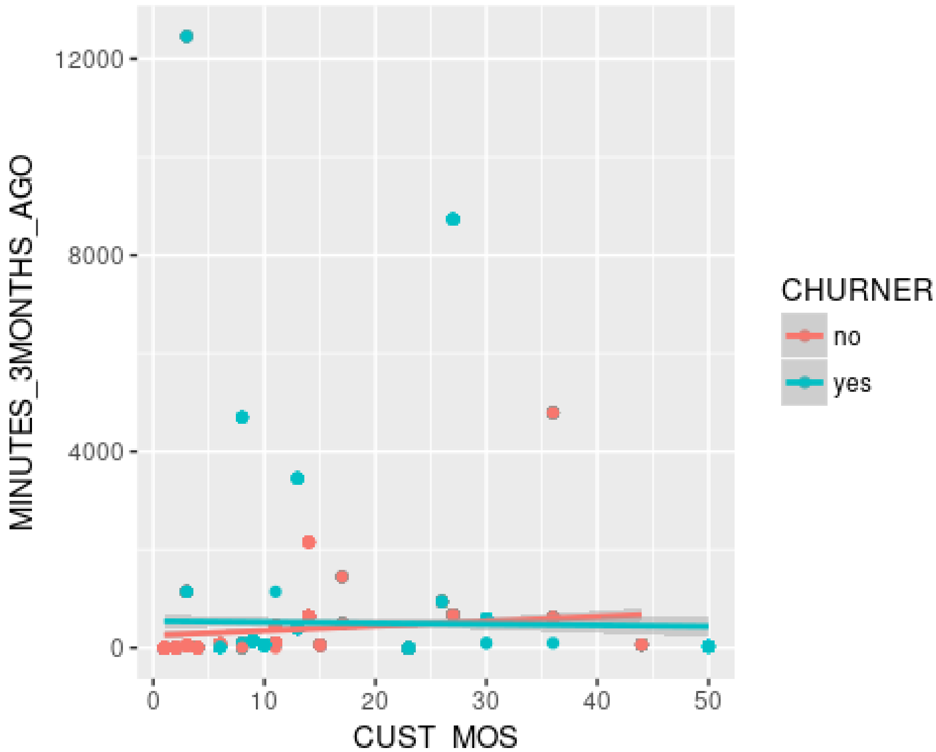


Correlation

```
cor(x=churn_data$TOT_MINUTES_USAGE, y=churn_data$MINUTES_PREV_MONTH,
use="all.obs", method="pearson")

## [1] 0.9568583

# http://www.statmethods.net/stats/correlations.html
```

1.7.8 CUST_MOS vs MINUTES_3MONTHS_AGO

```
ggplot(churn_data, aes(x=CUST_MOS, y=MINUTES_3MONTHS_AGO, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

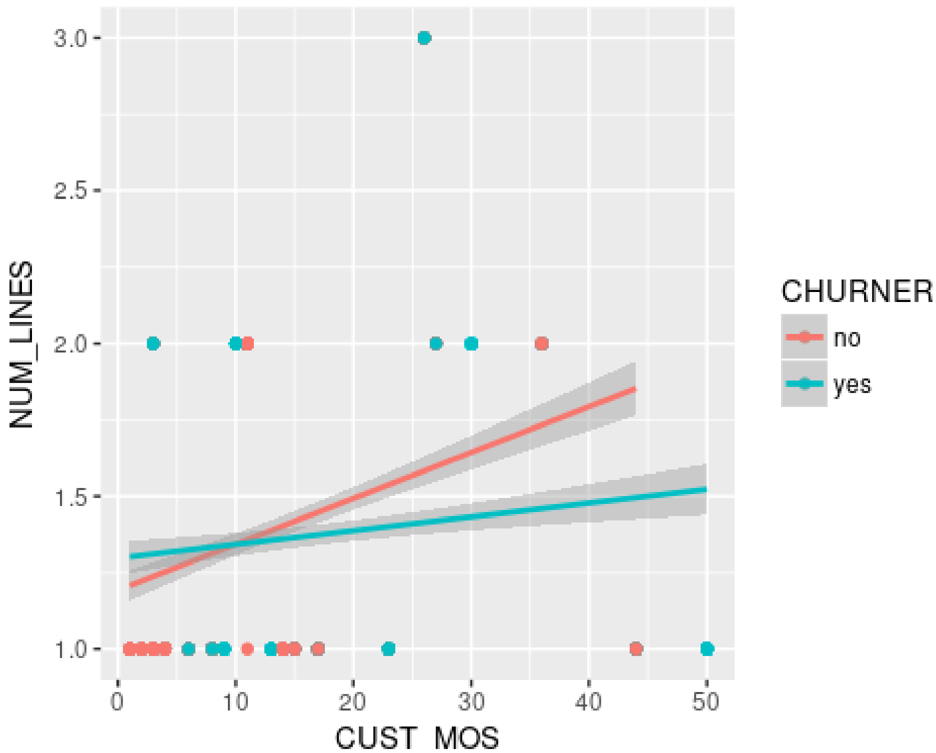


Correlation

```
cor(x=churn_data$CUST_MOS, y=churn_data$MINUTES_3MONTHS_AGO, use="all.obs",
method="pearson")

## [1] 0.03824096

# http://www.statmethods.net/stats/correlations.html
```

1.7.9 CUST_MOS vs NUM_LINES

```
ggplot(churn_data, aes(x=CUST_MOS, y=NUM_LINES, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

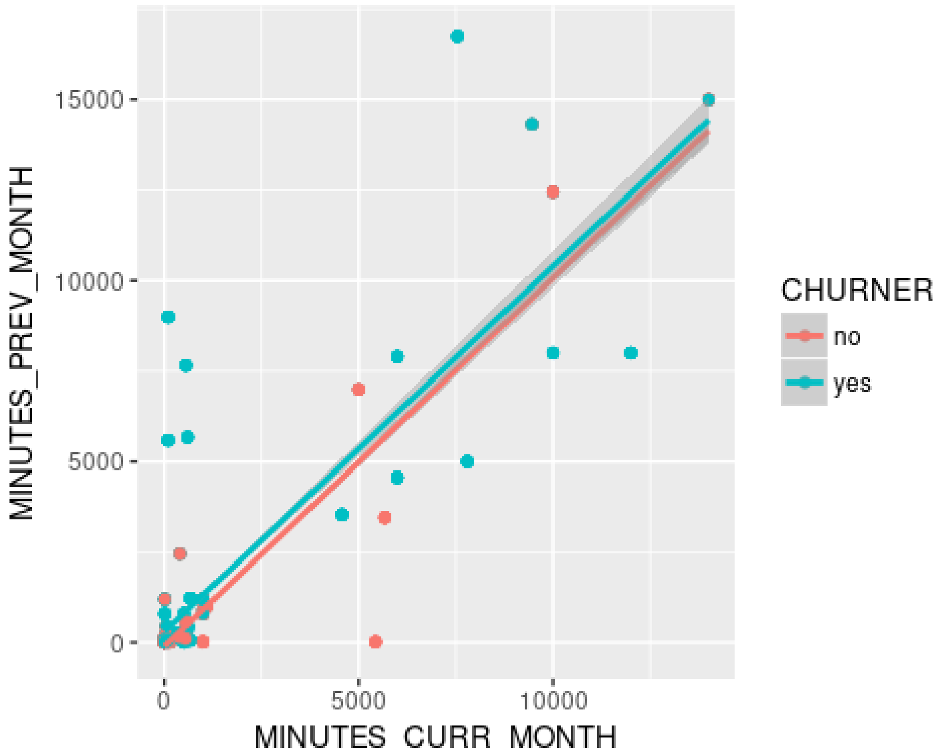


Correlation

```
cor(x=churn_data$CUST_MOS, y=churn_data$NUM_LINES, use="all.obs",
method="pearson")

## [1] 0.2028409

# http://www.statmethods.net/stats/correlations.html
```

1.7.10 MINUTES_CURR_MONTH vs MINUTES_PREV_MONTH

```
ggplot(churn_data, aes(x=MINUTES_CURR_MONTH, y=MINUTES_PREV_MONTH, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

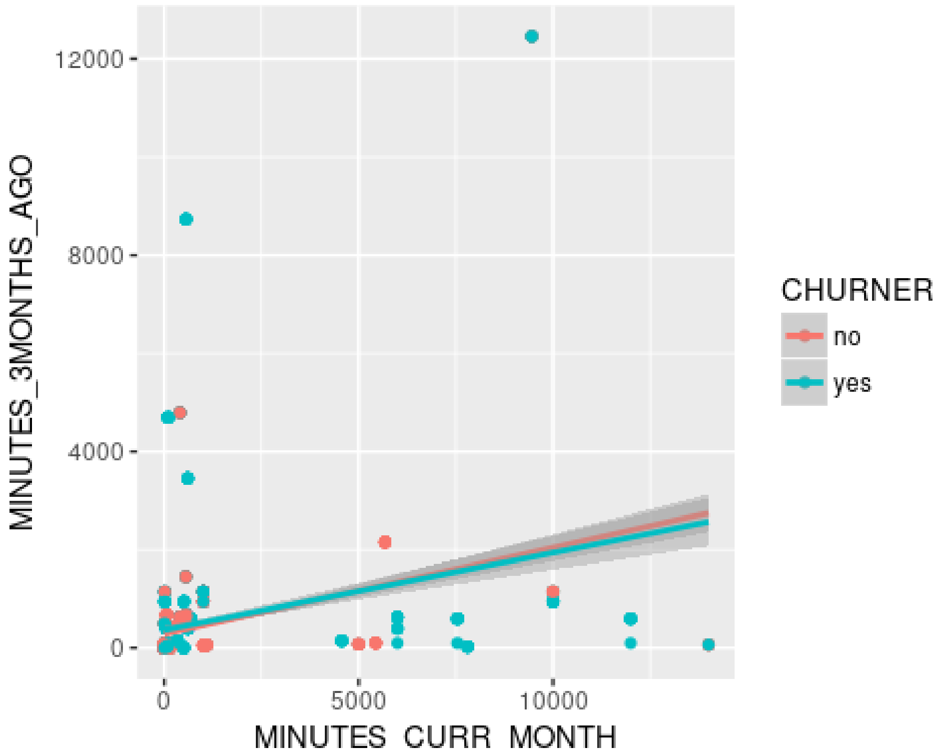


Correlation

```
cor(x=churn_data$MINUTES_CURR_MONTH, y=churn_data$MINUTES_PREV_MONTH,
use="all.obs", method="pearson")

## [1] 0.8332782

# http://www.statmethods.net/stats/correlations.html
```

1.7.11 MINUTES_CURR_MONTH vs MINUTES_3MONTHS_AGO

```
ggplot(churn_data, aes(x=MINUTES_CURR_MONTH, y=MINUTES_3MONTHS_AGO, col=CHURNER))+
    geom_point() +
    geom_smooth(method=lm)
```

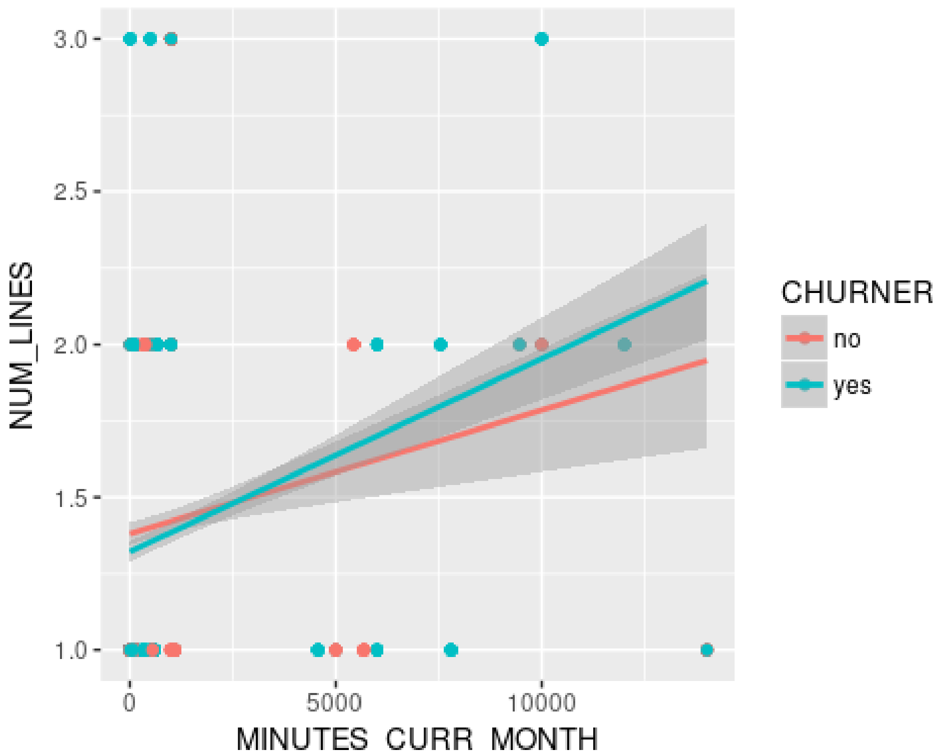


Correlation

```
cor(x=churn_data$MINUTES_CURR_MONTH, y=churn_data$MINUTES_3MONTHS_AGO,
use="all.obs", method="pearson")

## [1] 0.2806732

# http://www.statmethods.net/stats/correlations.html
```

1.7.12 MINUTES_CURR_MONTH vs NUM_LINES

```
ggplot(churn_data, aes(x=MINUTES_CURR_MONTH, y=NUM_LINES, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

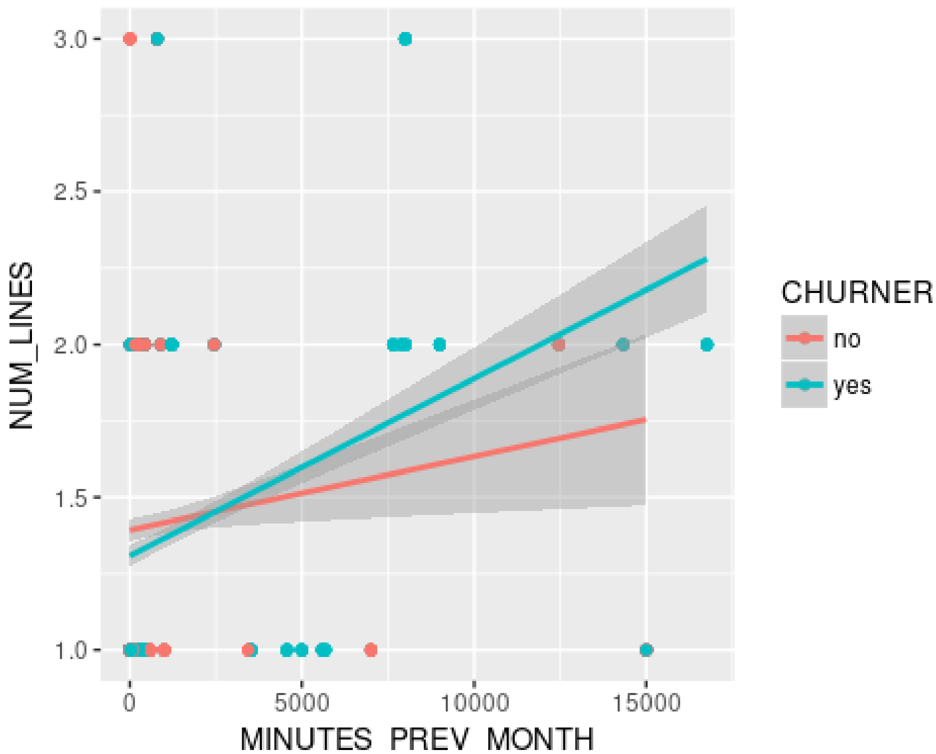


Correlation

```
cor(x=churn_data$MINUTES_CURR_MONTH, y=churn_data$NUM_LINES, use="all.obs", 
method="pearson")

## [1] 0.1932212

# http://www.statmethods.net/stats/correlations.html
```

1.7.13 MINUTES_PREV_MONTH vs NUM_LINES

```
ggplot(churn_data, aes(x=MINUTES_PREV_MONTH, y=NUM_LINES, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```

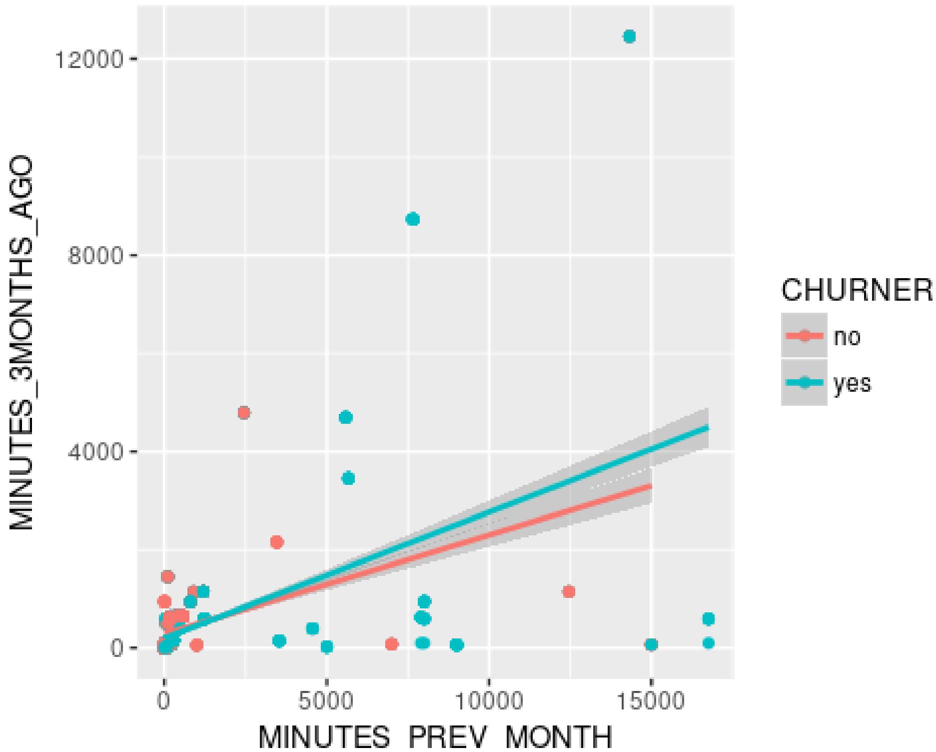


Correlation

```
cor(x=churn_data$MINUTES_PREV_MONTH, y=churn_data$NUM_LINES, use="all.obs",
method="pearson")

## [1] 0.2016433

# http://www.statmethods.net/stats/correlations.html
```

1.7.14 MINUTES_PREV_MONTH vs MINUTES_3MONTHS_AGO

```
ggplot(churn_data, aes(x=MINUTES_PREV_MONTH, y=MINUTES_3MONTHS_AGO, col=CHURNER))+
    geom_point() +
    geom_smooth(method=lm)
```

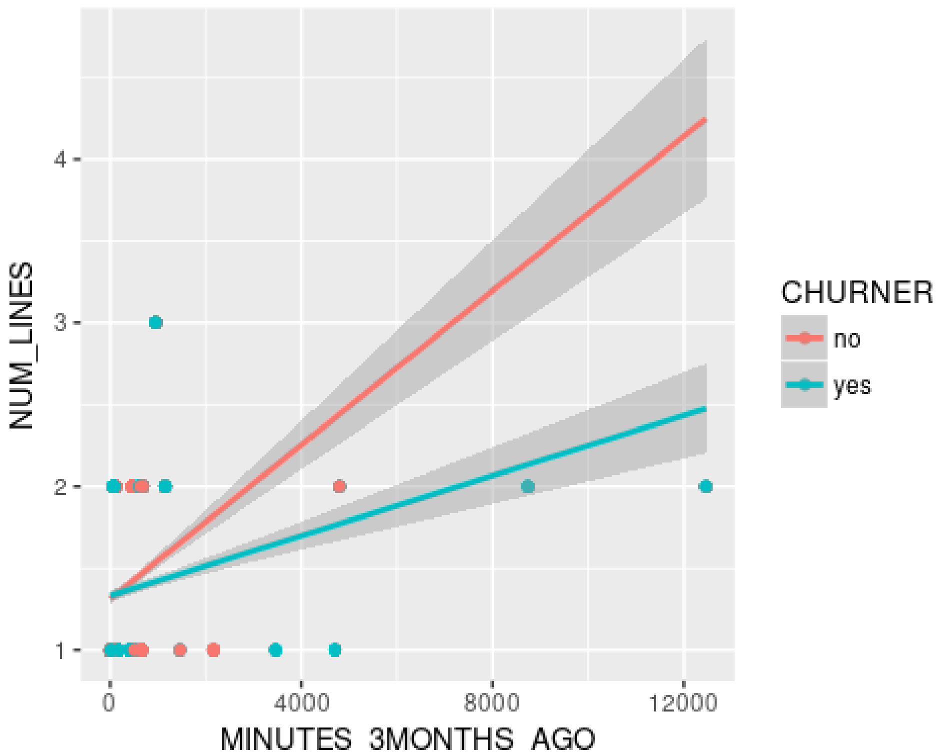


Correlation

```
cor(x=churn_data$MINUTES_PREV_MONTH, y=churn_data$MINUTES_3MONTHS_AGO,
use="all.obs", method="pearson")

## [1] 0.4989065

# http://www.statmethods.net/stats/correlations.html
```

1.7.15 MINUTES_3MONTHS_AGO vs NUM_LINES

```
ggplot(churn_data, aes(x=MINUTES_3MONTHS_AGO, y=NUM_LINES, col=CHURNER)) +
    geom_point() +
    geom_smooth(method=lm)
```



Correlation

```
# http://www.statmethods.net/stats/correlations.html
cor(x=churn_data$MINUTES_3MONTHS_AGO, y=churn_data$NUM_LINES, use="all.obs",
method="pearson")

## [1] 0.2624494
```

```
write.csv(churn_data, file="./churn_data_preprocessed.csv")
```

2 Weka

2.1 Mining algorithms

2.1.1 J48

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: churn_data_preprocessed-weka.filters.unsupervised.attribute.Remove-R1-2,4-6-weka.filters.unsupervised.attribute.NumericToNominal-R1,3,4,6,7-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

Instances: 2070

Attributes: 14

AREA_CODE

CUST_MOS

LONGDIST_FLAG

CALLWAITING_FLAG

NUM_LINES

VOICEMAIL_FLAG

MOBILE_PLAN

CONVERGENT_BILLING

GENDER

INCOME

PHONE_PLAN

EDUCATION

TOT_MINUTES_USAGE

CHURNER

Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

J48 pruned tree

```
CUST_MOS <= 0.877551
|  CONVERGENT_BILLING = Yes
|  |  AREA_CODE = 10040
|  |  |  CUST_MOS <= 0.020408: no (90.0/24.0)
|  |  |  CUST_MOS > 0.020408
|  |  |  |  GENDER = M: no (50.0/24.0)
|  |  |  |  GENDER = F: yes (10.0/3.0)
|  |  AREA_CODE = 15563
|  |  |  PHONE_PLAN = International
|  |  |  |  INCOME = High Income
|  |  |  |  |  TOT_MINUTES_USAGE <= 0.049452: yes (10.0)
|  |  |  |  |  TOT_MINUTES_USAGE > 0.049452: no (60.0/10.0)
|  |  |  |  INCOME = Medium Income: no (90.0/38.0)
|  |  |  |  INCOME = Low Income: yes (10.0)
|  |  |  PHONE_PLAN = National: yes (10.0)
|  |  |  PHONE_PLAN = Promo_plan: no (0.0)
|  |  |  PHONE_PLAN = Euro-Zone: no (0.0)
|  |  AREA_CODE = 21750: yes (200.0)
|  |  AREA_CODE = 36785: yes (120.0)
|  |  AREA_CODE = 45987: yes (90.0)
|  |  AREA_CODE = 55166: no (90.0)
|  CONVERGENT_BILLING = No
|  |  EDUCATION = Masters: no (120.0/32.0)
|  |  EDUCATION = Bachelors
|  |  |  VOICEMAIL_FLAG = 0: yes (90.0)
|  |  |  VOICEMAIL_FLAG = 1: no (111.0/41.0)
```

```
|   |   EDUCATION = High School: no (2.0)
|   |   EDUCATION = Post Primary
|   |   |   CUST_MOS <= 0.326531: no (427.0/125.0)
|   |   |   CUST_MOS > 0.326531
|   |   |   |   AREA_CODE = 10040: yes (0.0)
|   |   |   |   AREA_CODE = 15563: yes (0.0)
|   |   |   |   AREA_CODE = 21750: yes (70.0/10.0)
|   |   |   |   AREA_CODE = 36785: no (50.0/13.0)
|   |   |   |   AREA_CODE = 45987: yes (0.0)
|   |   |   |   AREA_CODE = 55166: yes (0.0)
|   |   EDUCATION = PhD: no (280.0/56.0)
|   |   EDUCATION = Primary: no (0.0)
CUST_MOS > 0.877551: yes (90.0)


Number of Leaves  :     28
Size of the tree :      40
Time taken to build model: 0.02 seconds
=== Evaluation on test split ===
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances         565               80.2557 %
Incorrectly Classified Instances       139               19.7443 %
Kappa statistic                          0.6099
Mean absolute error                      0.2643
Root mean squared error                  0.3802
Relative absolute error                 52.8582 %
```

Root relative squared error 76.0351 %

Total Number of Instances 704

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.651	0.033	0.956	0.651	0.775	0.646	0.843	0.872	yes
	0.967	0.349	0.718	0.967	0.824	0.646	0.843	0.785	no
Weighted Avg.	0.803	0.184	0.842	0.803	0.798	0.646	0.843	0.830	

=== Confusion Matrix ===

```
  a   b   <-- classified as
 239 128 |   a = yes
  11 326 |   b = no
```

2.1.1 jRip

Scheme: weka.classifiers.rules.JRip -F 3 -N 2.0 -O 2 -S 1

Relation: churn_data_preprocessed-weka.filters.unsupervised.attribute.Remove-R1-2,4-6-weka.filters.unsupervised.attribute.NumericToNominal-R1,3,4,6,7-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

Instances: 2070

Attributes: 14

AREA_CODE

CUST_MOS

LONGDIST_FLAG

CALLWAITING_FLAG

NUM_LINES

VOICEMAIL_FLAG

MOBILE_PLAN

CONVERGENT_BILLING

GENDER

INCOME

PHONE_PLAN

EDUCATION

TOT_MINUTES_USAGE

CHURNER

Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

JRIP rules:

===========

(CONVERGENT_BILLING = No) and (TOT_MINUTES_USAGE >= 0.00436) => CHURNER=no (690.0/200.0)

(AREA_CODE = 55166) => CHURNER=no (90.0/0.0)

(CUST_MOS <= 0.061224) => CHURNER=no (397.0/122.0)

(TOT_MINUTES_USAGE <= 0.003698) and (CUST_MOS <= 0.204082) => CHURNER=no (82.0/28.0)

(EDUCATION = PhD) and (TOT_MINUTES_USAGE <= 0.092281) and (INCOME = High Income) and (TOT_MINUTES_USAGE >= 0.053868) => CHURNER=no (60.0/10.0)

(EDUCATION = PhD) and (AREA_CODE = 45987) => CHURNER=no (70.0/19.0)

=> CHURNER=yes (681.0/10.0)

Number of Rules : 7

Time taken to build model: 0.23 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correctly Classified Instances	561	79.6875 %
Incorrectly Classified Instances	143	20.3125 %
Kappa statistic	0.599	
Mean absolute error	0.2773	
Root mean squared error	0.3848	
Relative absolute error	55.4645 %	
Root relative squared error	76.9526 %	
Total Number of Instances	704	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.638	0.030	0.959	0.638	0.766	0.638	0.809	0.813	yes
	0.970	0.362	0.711	0.970	0.821	0.638	0.809	0.707	no
Weighted Avg.	0.797	0.189	0.840	0.797	0.792	0.638	0.809	0.762	

=== Confusion Matrix ===

```
  a   b   <-- classified as
 234 133 |   a = yes
  10 327 |   b = no
```

2.1.1 PART

Scheme: weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1

Relation: churn_data_preprocessed-weka.filters.unsupervised.attribute.Remove-R1-2,4-6-weka.filters.unsupervised.attribute.NumericToNominal-R1,3,4,6,7-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

Instances: 2070

Attributes: 14

AREA_CODE

CUST_MOS

LONGDIST_FLAG

CALLWAITING_FLAG

NUM_LINES

VOICEMAIL_FLAG

MOBILE_PLAN

CONVERGENT_BILLING

GENDER

INCOME

PHONE_PLAN

EDUCATION

TOT_MINUTES_USAGE

CHURNER

Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

PART decision list

CUST_MOS <= 0.877551 AND CONVERGENT_BILLING = Yes AND AREA_CODE = 21750: yes (200.0)

CUST_MOS <= 0.877551 AND EDUCATION = Primary: yes (90.0)

CUST_MOS > 0.877551: yes (90.0)

TOT_MINUTES_USAGE > 0.22753 AND CONVERGENT_BILLING = Yes: yes (70.0/3.0)

PHONE_PLAN = National AND INCOME = Low Income: no (270.0/80.0)

EDUCATION = PhD AND TOT_MINUTES_USAGE > 0.002622 AND INCOME = High Income AND

TOT_MINUTES_USAGE <= 0.049452 AND MOBILE_PLAN = 0: no (170.0/48.0)

PHONE_PLAN = Promo_plan: no (260.0/44.0)

CUST_MOS <= 0.040816 AND AREA_CODE = 45987: no (89.0)

AREA_CODE = 45987 AND CUST_MOS <= 0.530612: yes (161.0/1.0)

INCOME = High Income AND EDUCATION = Post Primary: no (130.0/36.0)

INCOME = High Income AND TOT_MINUTES_USAGE > 0.049452: no (70.0/13.0)

CUST_MOS > 0.326531 AND PHONE_PLAN = International: yes (90.0/10.0)

INCOME = Medium Income AND GENDER = F: no (180.0/60.0)

INCOME = Medium Income AND PHONE_PLAN = International: no (90.0/38.0)

AREA_CODE = 36785: no (30.0/6.0)

PHONE_PLAN = National AND EDUCATION = PhD: no (20.0/9.0)

: no (60.0/29.0)

Number of Rules :   17

Time taken to build model: 0.21 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.02 seconds

=== Summary ===

Correctly Classified Instances 567 80.5398 %

Incorrectly Classified Instances 137 19.4602 %

Kappa statistic 0.6147

Mean absolute error 0.2566

Root mean squared error 0.3732

Relative absolute error 51.3241 %

Root relative squared error 74.6475 %

Total Number of Instances 704

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.678	0.056	0.929	0.678	0.784	0.640	0.847	0.870	yes
	0.944	0.322	0.729	0.944	0.823	0.640	0.847	0.787	no
Weighted Avg.	0.805	0.183	0.833	0.805	0.803	0.640	0.847	0.830	

=== Confusion Matrix ===

```
  a   b   <-- classified as
 249 118 |   a = yes
  19 318 |   b = no
```

2.1.1 ZeroR

Scheme: weka.classifiers.rules.ZeroR

Relation: churn_data_preprocessed-weka.filters.unsupervised.attribute.Remove-R1-2,4-6-weka.filters.unsupervised.attribute.NumericToNominal-R1,3,4,6,7-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

Instances: 2070

Attributes: 14

AREA_CODE

CUST_MOS

LONGDIST_FLAG

CALLWAITING_FLAG

NUM_LINES

VOICEMAIL_FLAG

MOBILE_PLAN

CONVERGENT_BILLING

GENDER

INCOME

PHONE_PLAN

EDUCATION

TOT_MINUTES_USAGE

CHURNER

Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

ZeroR predicts class value: yes

Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correctly Classified Instances 367 52.1307 %

Incorrectly Classified Instances 337 47.8693 %

Kappa statistic 0

Mean absolute error 0.5

Root mean squared error 0.5

Relative absolute error 100 %

Root relative squared error 100 %

Total Number of Instances 704

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	1.000	0.521	1.000	0.685	0.000	0.500	0.521	yes
	0.000	0.000	0.000	0.000	0.000	0.000	0.500	0.479	no
Weighted Avg.	0.521	0.521	0.272	0.521	0.357	0.000	0.500	0.501	

=== Confusion Matrix ===

```
  a   b   <-- classified as
 367   0 |   a = yes
 337   0 |   b = no
```

Bibliography

Your Bibliography: Gim.unmc.edu. (2017). The Area Under an ROC Curve. [online] Available at: http://gim.unmc.edu/dxtests/roc3.htm [Accessed 26 Mar. 2017].