R Notebook

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 EDUCATION   
## 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

# Preprocess the data

## 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

## Remove the duplicates

churn\_data <- churn\_data[!duplicated(churn\_data$CUST\_ID), ]

nrow(churn\_data)

## [1] 2070

## 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)

## Addressing the issues of the missing values.

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

### 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

### 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

### 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

### 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

### 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

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

# Predictor Information

## CUST\_ID

Unique Account Number that identifies an account holder

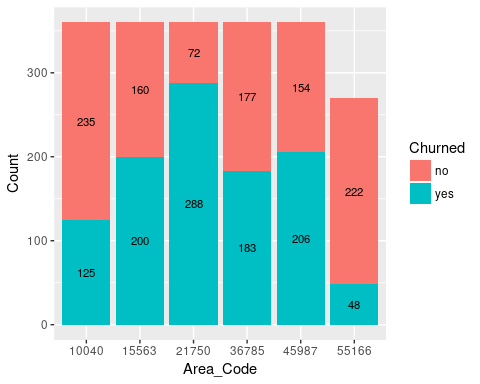
|  |  |
| --- | --- |
| Predictor | CUST\_ID |
| attribute type | Nominal ` |
| Max | 2070 |
| Min | 1 |

## AREA\_CODE

Geographical area account holder resides

|  |  |
| --- | --- |
| Predictor | AREA\_CODE |
| attribute type | Nominal |
| %Missing Values | 0% |
| Mode | 10040, 15563, 21750, 36785, 45987 |

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))

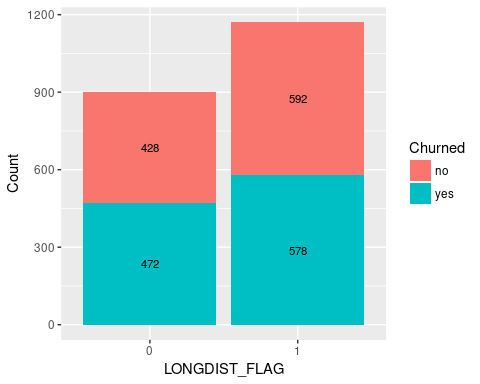


## LONGDIST\_FLAG

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

|  |  |
| --- | --- |
| Predictor | LONGDIST\_FLAG |
| attribute type | Nominal |
| %Missing Values | 0 |
| Mode | 1 |

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))

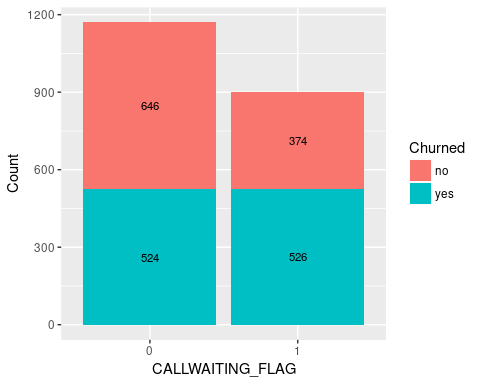


## CALLWAITING\_FLAG

Whether the customer has call waiting

|  |  |
| --- | --- |
| Predictor | CALLWAITING\_FLAG |
| attribute type | Nominal |
| %Missing Values | 0% |
| Mode | 0 |

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))

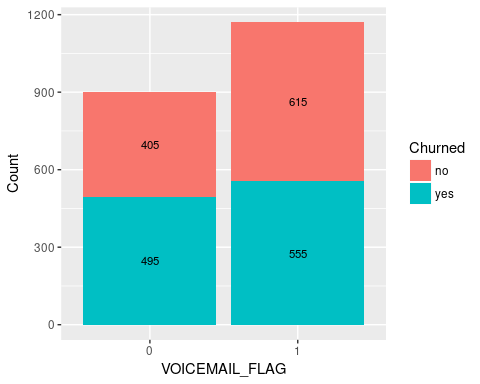


## VOICEMAIL\_FLAG

Whether the customer has voice mail

|  |  |
| --- | --- |
| Predictor | VOICEMAIL\_FLAG |
| attribute type | Nominal |
| %Missing Values | 0% |
| Mode | 1 |

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))

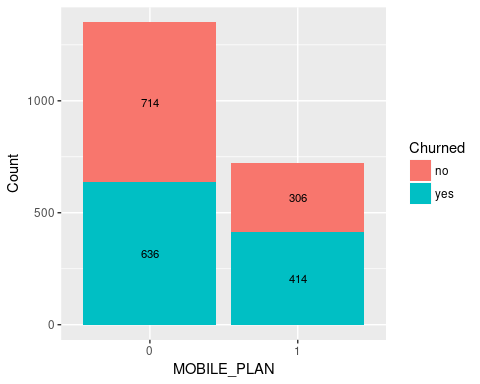


## MOBILE\_PLAN

Has the customer signed up to the mobile phone plan

|  |  |
| --- | --- |
| Predictor | MOBILE\_PLAN |
| attribute type | Nominal |
| %Missing Values | 0% |
| Mode | 0 |

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))

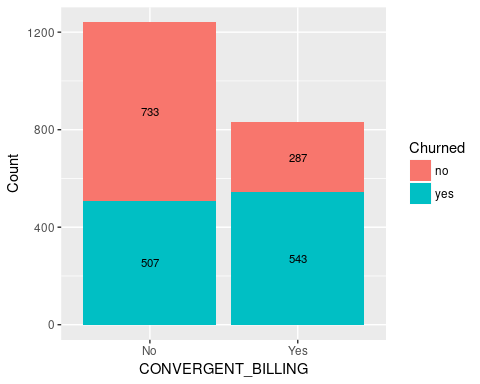


## CONVERGENT\_BILLING

All service charges consolidated onto one bill

|  |  |
| --- | --- |
| Predictor | CONVERGENT\_BILLING |
| attribute type | Nominal |
| %Missing Values | 0% |
| Mode | No |

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))

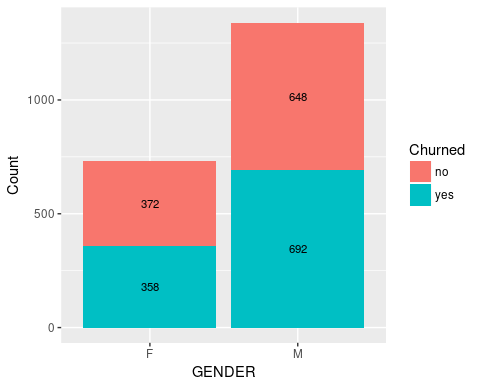


## GENDER

Account holder's gender

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

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))



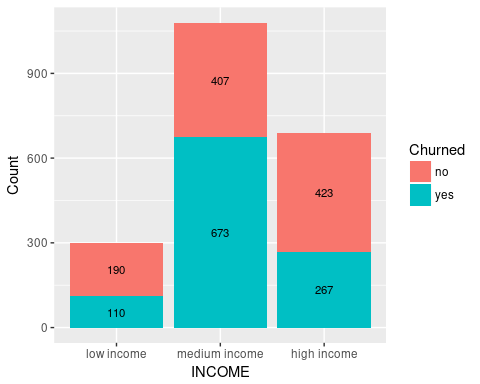
## INCOME

Account holder's annual income (€)

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

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

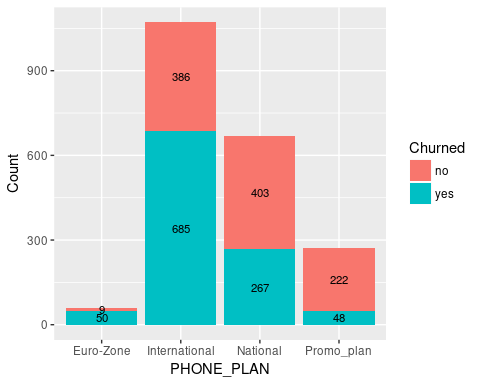


## PHONE\_PLAN

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% |
| Mode | International |

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))

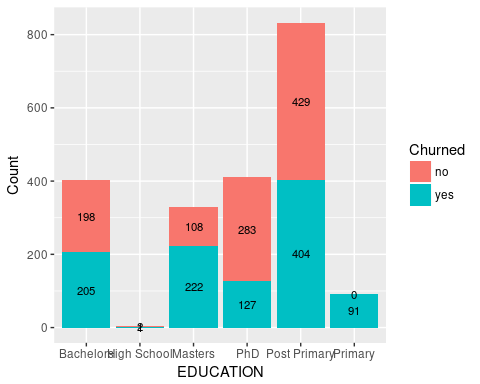


## EDUCATION

Highest Level of education attainment the account holder has achieved

|  |  |
| --- | --- |
| Predictor | EDUCATION |
| attribute type | Nominal |
| %Missing Values | 0% |
| Mode | Post Primary |

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))



## 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 | 748.0917874 |
| Mode | 2, 5 |
| median | 105 |
| Standard deviation | 2017.5556123 |

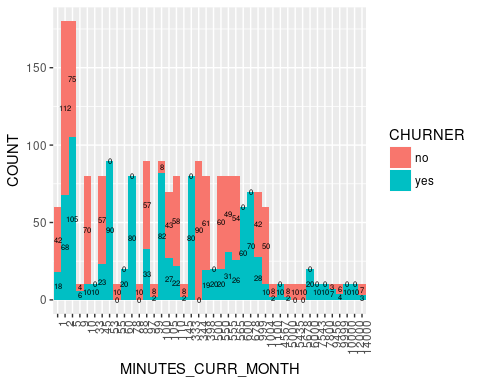
MINUTES\_CURR\_MONTH Skewness

(3 \* (mean(churn\_data$MINUTES\_PREV\_MONTH) - median(churn\_data$MINUTES\_PREV\_MONTH)))/ sd(churn\_data$MINUTES\_PREV\_MONTH)

## [1] 0.9312979

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



## MINUTES\_PREV\_MONTH

Phone Minutes used in the previous month

|  |  |
| --- | --- |
| Predictor | MINUTES\_PREV\_MONTH |
| attribute type | Numerical |
| %Missing Values |  |
| Max | 16754 |
| Min | 0 |
| Mean | 864.3188406 |
| Mode | 0 |
| median | 98 |

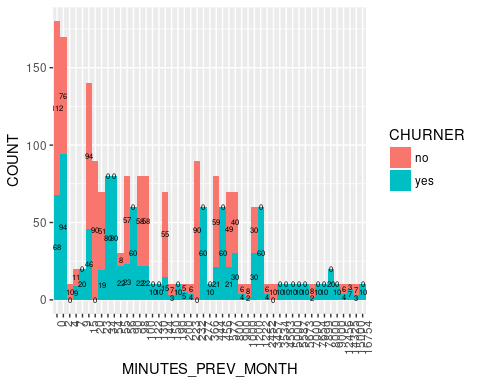
MINUTES\_PREV\_MONTH Skewness

(3 \* (mean(churn\_data$MINUTES\_PREV\_MONTH) - median(churn\_data$MINUTES\_PREV\_MONTH)))/ sd(churn\_data$MINUTES\_PREV\_MONTH)

## [1] 0.9312979

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

 ## MINUTES\_3MONTHS\_AGO

Phone Minutes used in the 3 month PREVIOUS

|  |  |
| --- | --- |
| Predictor | MINUTES\_3MONTHS\_AGO |
| attribute type | Numerical |
| %Missing Values |  |
| Max | 12456 |
| Min | 0 |
| Mean | 452.7463768 |
| Mode | 0 |
| median | 97 |

MINUTES\_3MONTHS\_AGO Skewness

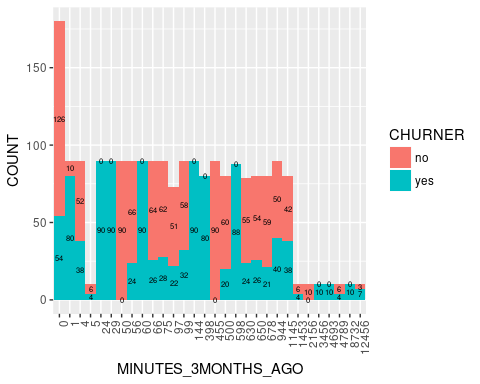
Right-skewness data – Is positive, as mean is greater than the median

(3 \* (mean(churn\_data$MINUTES\_3MONTHS\_AGO) - median(churn\_data$MINUTES\_3MONTHS\_AGO)))/ sd(churn\_data$MINUTES\_3MONTHS\_AGO)

## [1] 0.9012361

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



## CUST\_MOS

The number of continuous months the Customer is with the provider

|  |  |
| --- | --- |
| Predictor | CUST\_MOS |
| attribute type | Numerical |
| %Missing Values | 0 |
| Max | 50 |
| Min | 1 |
| Mean | 16.0545894 |
| Mode | 11 |
| median | 11 |

CUST\_MOS Skewness

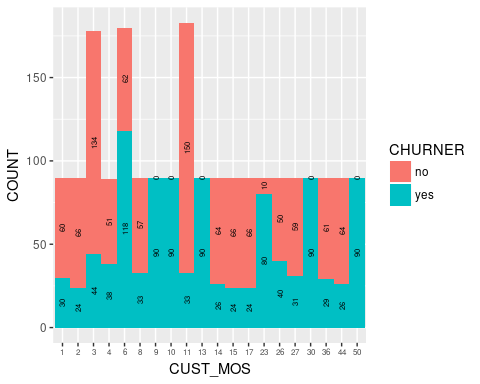
Right-skewness data – Is positive, as mean is greater than the median

(3 \* (mean(churn\_data$CUST\_MOS) - median(churn\_data$CUST\_MOS)))/ sd(churn\_data$CUST\_MOS)

## [1] 1.132926

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



## TOT\_MINUTES\_USAGE

The total number of minutes used to date

|  |  |
| --- | --- |
| Predictor | TOT\_MINUTES\_USAGE |
| attribute type | Numeric |
| %Missing Values | 0% |
| Max | 3.623710^{4} |
| Min | 0 |
| Mean | 2037.0932367 |
| Mode | 0 |
| median | 264 |
| Standard deviation | 4883.9078032 |

TOT\_MINUTES\_USAGE Skewness

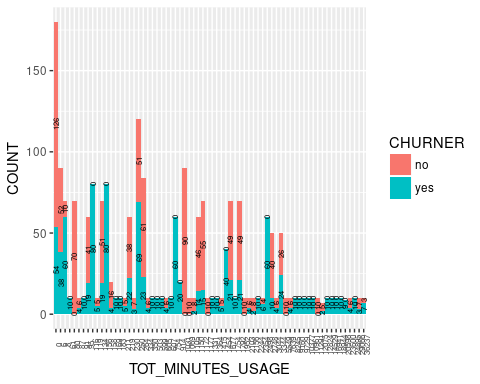
Right-skewness data – Is positive, as mean is greater than the median

(3 \* (mean(churn\_data$TOT\_MINUTES\_USAGE) - median(churn\_data$TOT\_MINUTES\_USAGE)))/ sd(churn\_data$TOT\_MINUTES\_USAGE)

## [1] 1.089144

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



## 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

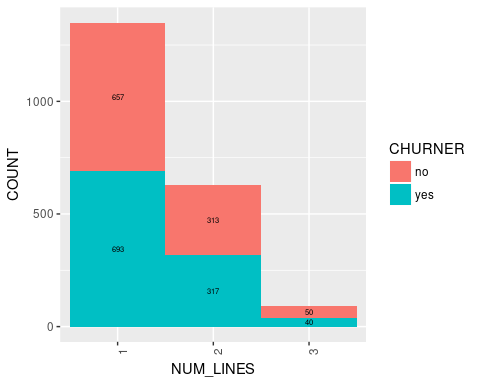
Right-skewness data – Is positive, as mean is greater than the median

(3 \* (mean(churn\_data$NUM\_LINES) - median(churn\_data$NUM\_LINES)))/ sd(churn\_data$NUM\_LINES)

## [1] 2.058233

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



# 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

(3 \* (mean(churn\_data$MINUTES\_3MONTHS\_AGO) - median(churn\_data$MINUTES\_3MONTHS\_AGO)))/ sd(churn\_data$MINUTES\_3MONTHS\_AGO)

## [1] 0.9012361

(3 \* (mean(churn\_data$MINUTES\_PREV\_MONTH) - median(churn\_data$MINUTES\_PREV\_MONTH)))/ sd(churn\_data$MINUTES\_PREV\_MONTH)

## [1] 0.9312979

(3 \* (mean(churn\_data$MINUTES\_CURR\_MONTH) - median(churn\_data$MINUTES\_CURR\_MONTH)))/ sd(churn\_data$MINUTES\_CURR\_MONTH)

## [1] 0.956244

(3 \* (mean(churn\_data$TOT\_MINUTES\_USAGE) - median(churn\_data$TOT\_MINUTES\_USAGE)))/ sd(churn\_data$TOT\_MINUTES\_USAGE)

## [1] 1.089144

(3 \* (mean(churn\_data$NUM\_LINES) - median(churn\_data$NUM\_LINES)))/ sd(churn\_data$NUM\_LINES)

## [1] 2.058233

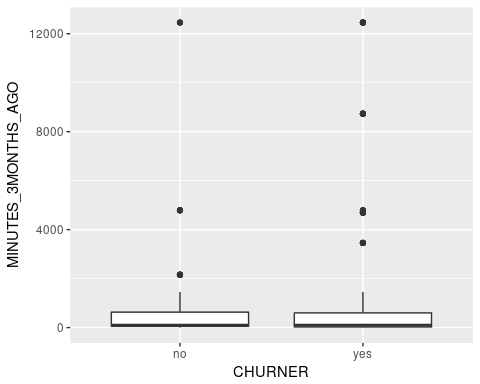
(3 \* (mean(churn\_data$CUST\_MOS) - median(churn\_data$CUST\_MOS)))/ sd(churn\_data$CUST\_MOS)

## [1] 1.132926

# Graphical methods to identify outliers

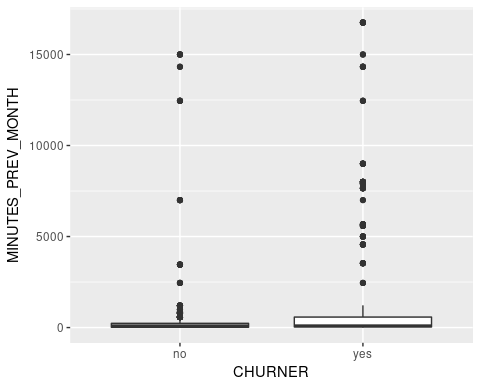
## MINUTES\_3MONTHS\_AGO

ggplot(data=churn\_data, aes(x=CHURNER, y=MINUTES\_3MONTHS\_AGO))+ geom\_boxplot()



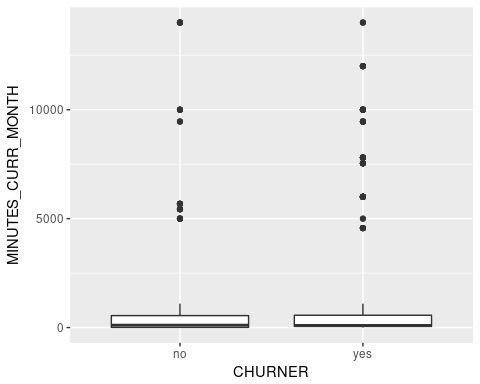
## MINUTES\_PREV\_MONTH

ggplot(data=churn\_data, aes(x=CHURNER, y=MINUTES\_PREV\_MONTH))+ geom\_boxplot()



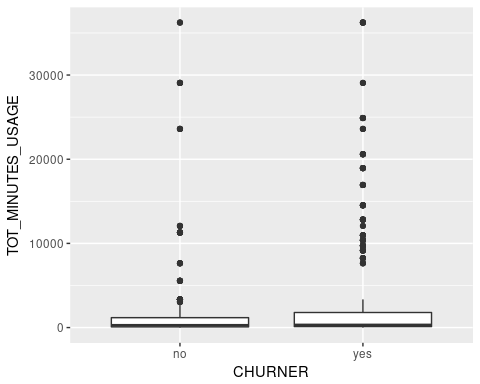
## MINUTES\_CURR\_MONTH

ggplot(data=churn\_data, aes(x=CHURNER, y=MINUTES\_CURR\_MONTH))+ geom\_boxplot()



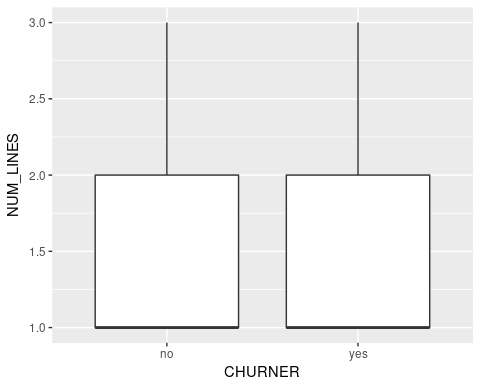
## TOT\_MINUTES\_USAGE

ggplot(data=churn\_data, aes(x=CHURNER, y=TOT\_MINUTES\_USAGE))+ geom\_boxplot()



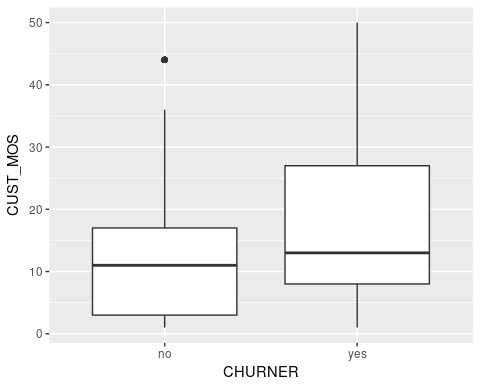
## NUM\_LINES

ggplot(data=churn\_data, aes(x=CHURNER, y=NUM\_LINES))+ geom\_boxplot()



## CUST\_MOS

ggplot(data=churn\_data, aes(x=CHURNER, y=CUST\_MOS))+ geom\_boxplot()



# 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

# 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

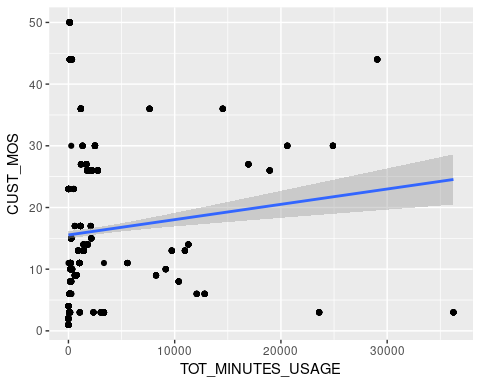
## [1] 1.089144

# 2D Scatter plots to investigate correlation

2D Scatter plots to investigate correlation between numeric variables

## TOT\_MINUTES\_USAGE vs CUST\_MOS

ggplot(churn\_data, aes(x=TOT\_MINUTES\_USAGE, y=CUST\_MOS)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

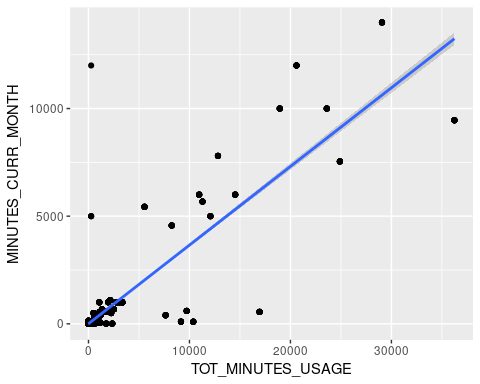
cor(x=churn\_data$TOT\_MINUTES\_USAGE, y=churn\_data$CUST\_MOS, use="all.obs", method="pearson")

## [1] 0.09055857

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

## TOT\_MINUTES\_USAGE vs MINUTES\_CURR\_MONTH

ggplot(churn\_data, aes(x=TOT\_MINUTES\_USAGE, y=MINUTES\_CURR\_MONTH)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

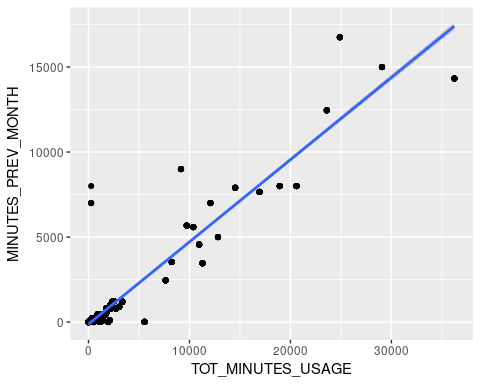
cor(x=churn\_data$TOT\_MINUTES\_USAGE, y=churn\_data$MINUTES\_CURR\_MONTH, use="all.obs", method="pearson")

## [1] 0.8844237

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

## TOT\_MINUTES\_USAGE vs MINUTES\_PREV\_MONTH

ggplot(churn\_data, aes(x=TOT\_MINUTES\_USAGE, y=MINUTES\_PREV\_MONTH)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

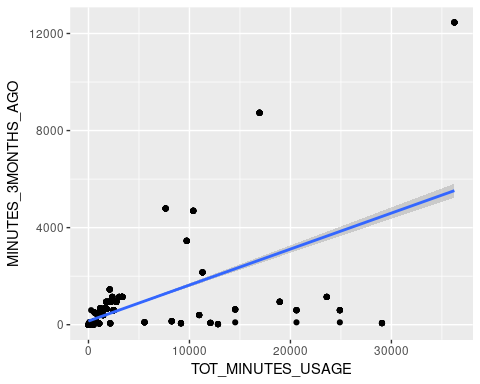
cor(x=churn\_data$TOT\_MINUTES\_USAGE, y=churn\_data$MINUTES\_PREV\_MONTH, use="all.obs", method="pearson")

## [1] 0.9568583

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

## TOT\_MINUTES\_USAGE vs MINUTES\_3MONTHS\_AGO

ggplot(churn\_data, aes(x=TOT\_MINUTES\_USAGE, y=MINUTES\_3MONTHS\_AGO)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

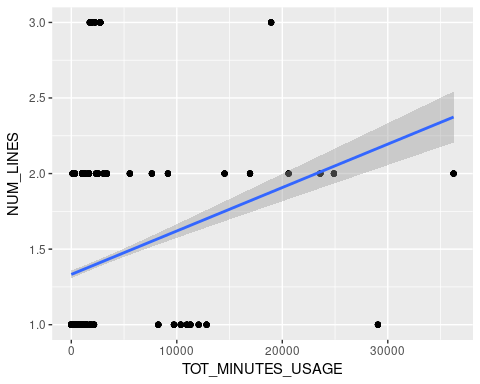
cor(x=churn\_data$TOT\_MINUTES\_USAGE, y=churn\_data$MINUTES\_3MONTHS\_AGO, use="all.obs", method="pearson")

## [1] 0.611558

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

## TOT\_MINUTES\_USAGE vs NUM\_LINES

ggplot(churn\_data, aes(x=TOT\_MINUTES\_USAGE, y=NUM\_LINES)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

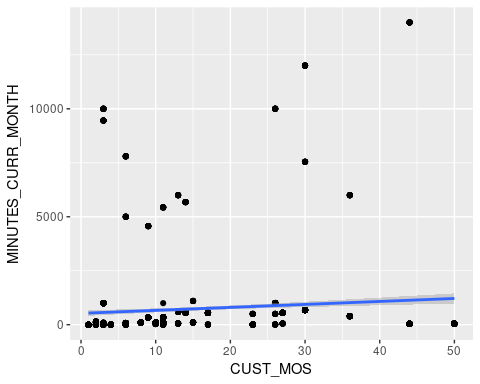
cor(x=churn\_data$TOT\_MINUTES\_USAGE, y=churn\_data$NUM\_LINES, use="all.obs", method="pearson")

## [1] 0.2460581

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

## CUST\_MOS vs MINUTES\_CURR\_MONTH

ggplot(churn\_data, aes(x=CUST\_MOS, y=MINUTES\_CURR\_MONTH)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

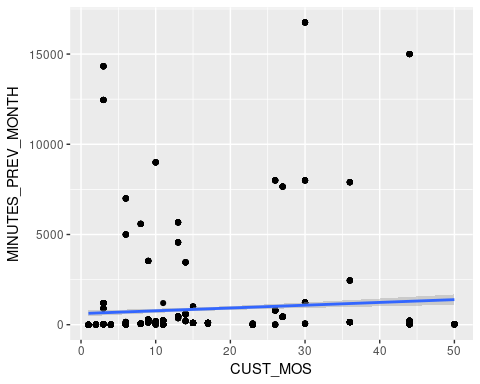
cor(x=churn\_data$TOT\_MINUTES\_USAGE, y=churn\_data$MINUTES\_CURR\_MONTH, use="all.obs", method="pearson")

## [1] 0.8844237

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

## CUST\_MOS vs MINUTES\_PREV\_MONTH

ggplot(churn\_data, aes(x=CUST\_MOS, y=MINUTES\_PREV\_MONTH)) +  
 geom\_point() +  
 geom\_smooth(method=lm)

 Correlation

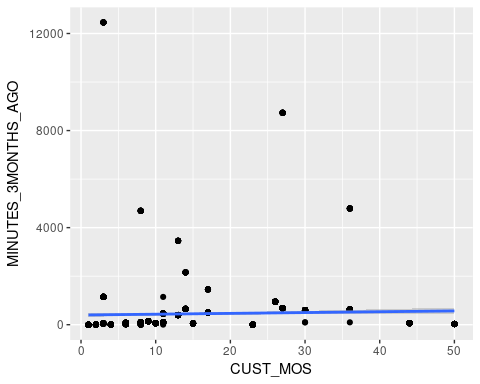
cor(x=churn\_data$TOT\_MINUTES\_USAGE, y=churn\_data$MINUTES\_PREV\_MONTH, use="all.obs", method="pearson")

## [1] 0.9568583

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

## CUST\_MOS vs MINUTES\_3MONTHS\_AGO

ggplot(churn\_data, aes(x=CUST\_MOS, y=MINUTES\_3MONTHS\_AGO)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

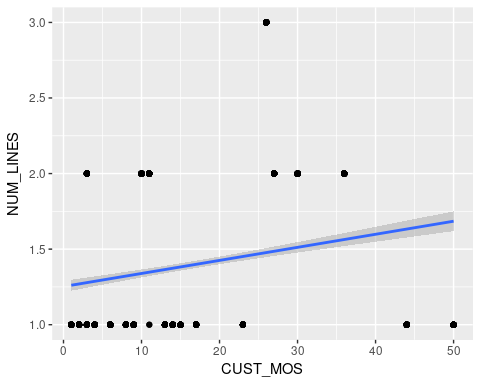
cor(x=churn\_data$CUST\_MOS, y=churn\_data$MINUTES\_3MONTHS\_AGO, use="all.obs", method="pearson")

## [1] 0.03824096

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

## CUST\_MOS vs NUM\_LINES

ggplot(churn\_data, aes(x=CUST\_MOS, y=NUM\_LINES)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

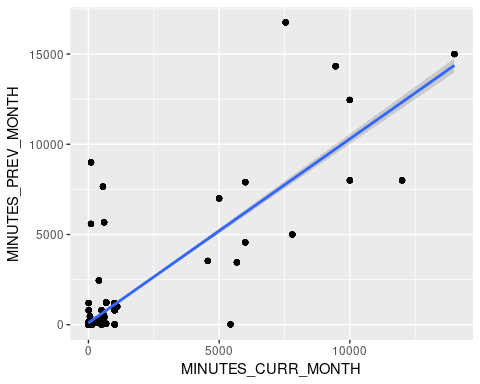
cor(x=churn\_data$CUST\_MOS, y=churn\_data$NUM\_LINES, use="all.obs", method="pearson")

## [1] 0.2028409

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

## MINUTES\_CURR\_MONTH vs MINUTES\_PREV\_MONTH

ggplot(churn\_data, aes(x=MINUTES\_CURR\_MONTH, y=MINUTES\_PREV\_MONTH)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

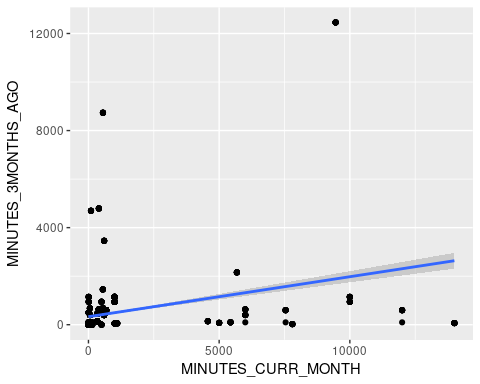
cor(x=churn\_data$MINUTES\_CURR\_MONTH, y=churn\_data$MINUTES\_PREV\_MONTH, use="all.obs", method="pearson")

## [1] 0.8332782

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

## MINUTES\_CURR\_MONTH vs MINUTES\_3MONTHS\_AGO

ggplot(churn\_data, aes(x=MINUTES\_CURR\_MONTH, y=MINUTES\_3MONTHS\_AGO)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

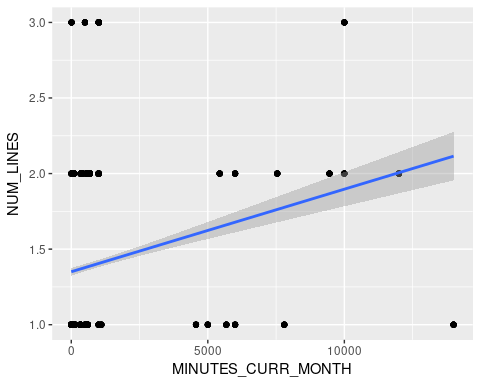
cor(x=churn\_data$MINUTES\_CURR\_MONTH, y=churn\_data$MINUTES\_3MONTHS\_AGO, use="all.obs", method="pearson")

## [1] 0.2806732

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

## MINUTES\_CURR\_MONTH vs NUM\_LINES

ggplot(churn\_data, aes(x=MINUTES\_CURR\_MONTH, y=NUM\_LINES)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

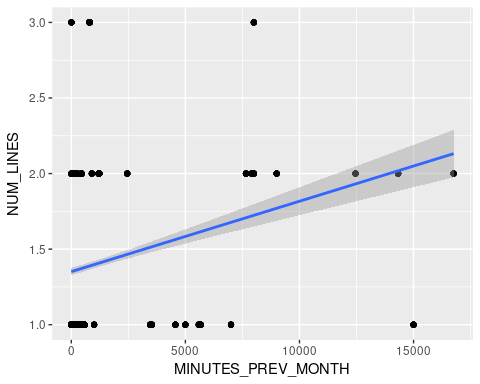
cor(x=churn\_data$MINUTES\_CURR\_MONTH, y=churn\_data$NUM\_LINES, use="all.obs", method="pearson")

## [1] 0.1932212

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

## MINUTES\_PREV\_MONTH vs NUM\_LINES

ggplot(churn\_data, aes(x=MINUTES\_PREV\_MONTH, y=NUM\_LINES)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

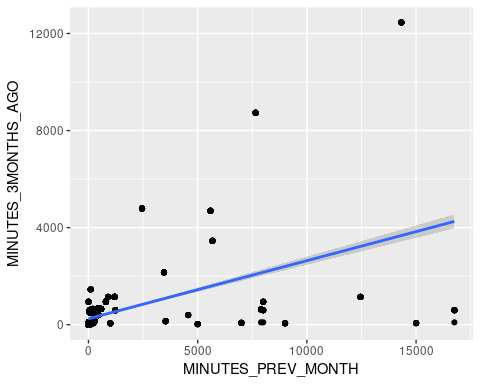
cor(x=churn\_data$MINUTES\_PREV\_MONTH, y=churn\_data$NUM\_LINES, use="all.obs", method="pearson")

## [1] 0.2016433

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

## MINUTES\_PREV\_MONTH vs MINUTES\_3MONTHS\_AGO

ggplot(churn\_data, aes(x=MINUTES\_PREV\_MONTH, y=MINUTES\_3MONTHS\_AGO)) +  
 geom\_point() +  
 geom\_smooth(method=lm)



Correlation

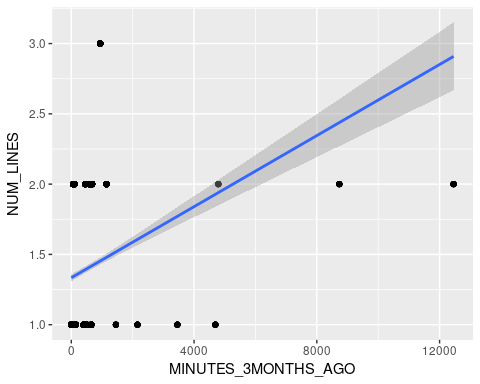
cor(x=churn\_data$MINUTES\_PREV\_MONTH, y=churn\_data$MINUTES\_3MONTHS\_AGO, use="all.obs", method="pearson")

## [1] 0.4989065

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

## MINUTES\_3MONTHS\_AGO vs NUM\_LINES

ggplot(churn\_data, aes(x=MINUTES\_3MONTHS\_AGO, y=NUM\_LINES)) +  
 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")