Group\_1 Project

Abhinay Chiranjeeth Marneni & Meshari Alqeryan

2022-12-10

## loading library functions packages

library(dplyr)

library(pROC)

library(plyr)

library(ggplot2)  
library(caret)

library(party)

library(rpart)  
library(rpart.plot)

library(esquisse)  
library(readr)

#Loading the Churn file

Churn\_Train <- read\_csv("C:/Users/HUAWEI/Downloads/Churn\_Train.csv")

## Rows: 3333 Columns: 20  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): state, area\_code, international\_plan, voice\_mail\_plan, churn  
## dbl (15): account\_length, number\_vmail\_messages, total\_day\_minutes, total\_da...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

View(Churn\_Train)  
data = Churn\_Train  
summary(data)

## state account\_length area\_code international\_plan  
## Length:3333 Min. :-209.00 Length:3333 Length:3333   
## Class :character 1st Qu.: 72.00 Class :character Class :character   
## Mode :character Median : 100.00 Mode :character Mode :character   
## Mean : 97.32   
## 3rd Qu.: 127.00   
## Max. : 243.00   
## NA's :501   
## voice\_mail\_plan number\_vmail\_messages total\_day\_minutes total\_day\_calls  
## Length:3333 Min. :-10.000 Min. : 0.0 Min. : 0.0   
## Class :character 1st Qu.: 0.000 1st Qu.: 149.3 1st Qu.: 87.0   
## Mode :character Median : 0.000 Median : 190.5 Median :101.0   
## Mean : 7.333 Mean : 418.9 Mean :100.3   
## 3rd Qu.: 16.000 3rd Qu.: 237.8 3rd Qu.:114.0   
## Max. : 51.000 Max. :2185.1 Max. :165.0   
## NA's :200 NA's :200 NA's :200   
## total\_day\_charge total\_eve\_minutes total\_eve\_calls total\_eve\_charge  
## Min. : 0.00 Min. : 0.0 Min. : 0.0 Min. : 0.00   
## 1st Qu.:24.45 1st Qu.: 170.5 1st Qu.: 87.0 1st Qu.:14.14   
## Median :30.65 Median : 209.9 Median :100.0 Median :17.09   
## Mean :30.63 Mean : 324.3 Mean :100.1 Mean :17.08   
## 3rd Qu.:36.84 3rd Qu.: 257.6 3rd Qu.:114.0 3rd Qu.:20.00   
## Max. :59.64 Max. :1244.2 Max. :170.0 Max. :30.91   
## NA's :200 NA's :301 NA's :200 NA's :200   
## total\_night\_minutes total\_night\_calls total\_night\_charge total\_intl\_minutes  
## Min. : 23.2 Min. : 33.0 Min. : 1.040 Min. : 0.00   
## 1st Qu.:167.3 1st Qu.: 87.0 1st Qu.: 7.530 1st Qu.: 8.50   
## Median :201.4 Median :100.0 Median : 9.060 Median :10.30   
## Mean :201.2 Mean :100.1 Mean : 9.054 Mean :10.23   
## 3rd Qu.:235.3 3rd Qu.:113.0 3rd Qu.:10.590 3rd Qu.:12.10   
## Max. :395.0 Max. :175.0 Max. :17.770 Max. :20.00   
## NA's :200 NA's :200 NA's :200   
## total\_intl\_calls total\_intl\_charge number\_customer\_service\_calls  
## Min. : 0.00 Min. :0.000 Min. :0.000   
## 1st Qu.: 3.00 1st Qu.:2.300 1st Qu.:1.000   
## Median : 4.00 Median :2.780 Median :1.000   
## Mean : 4.47 Mean :2.762 Mean :1.561   
## 3rd Qu.: 6.00 3rd Qu.:3.270 3rd Qu.:2.000   
## Max. :20.00 Max. :5.400 Max. :9.000   
## NA's :301 NA's :200 NA's :200   
## churn   
## Length:3333   
## Class :character   
## Mode :character   
##   
##   
##   
##

# analysing count of NA value in the dataset

sapply(data, function(x) sum(is.na(x))) # NA data

## state account\_length   
## 0 501   
## area\_code international\_plan   
## 0 0   
## voice\_mail\_plan number\_vmail\_messages   
## 0 200   
## total\_day\_minutes total\_day\_calls   
## 200 200   
## total\_day\_charge total\_eve\_minutes   
## 200 301   
## total\_eve\_calls total\_eve\_charge   
## 200 200   
## total\_night\_minutes total\_night\_calls   
## 200 0   
## total\_night\_charge total\_intl\_minutes   
## 200 200   
## total\_intl\_calls total\_intl\_charge   
## 301 200   
## number\_customer\_service\_calls churn   
## 200 0

## removing NA values

data<- na.omit(data)

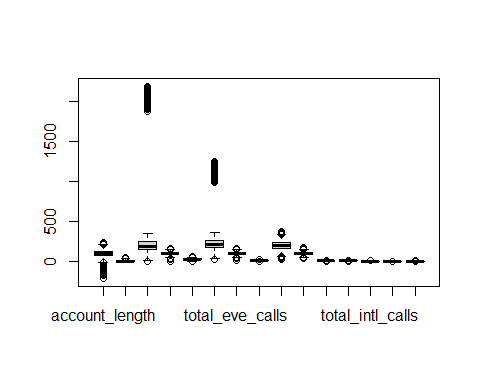
## removing Negative values

abs(data$account\_length + data$number\_vmail\_messages)

## [1] 125 108 82 83 135 28 86 125 115 131 65 143 93 33 122 141 155 184  
## [19] 160 54 113 73 142 64 228 55 22 70 177 49 116 141 167 87 184 80  
## [37] 80 91 18 46 52 91 113 36 132 137 103 116 111 160 78 44 164 115  
## [55] 128 43 161 185 131 114 132 139 133 164 138 157 115 105 80 105 78 125  
## [73] 57 83 208 122 118 74 81 135 159 167 154 152 114 103 95 133 105 37  
## [91] 129 94 98 161 78 45 113 77 176 120 142 77 70 95 131 170 76 108  
## [109] 64 150 97 112 73 25 39 162 60 39 113 110 93 157 125 68 80 86  
## [127] 99 66 138 129 165 80 86 146 130 41 145 179 98 115 128 172 106 57  
## [145] 127 118 40 3 221 144 72 134 112 151 39 176 56 52 154 143 78 34  
## [163] 166 93 80 118 151 100 84 180 79 74 60 104 183 139 122 97 129 67  
## [181] 93 166 85 62 114 175 129 107 92 138 141 110 97 65 82 123 115 81  
## [199] 122 59 185 65 169 45 127 128 98 152 113 165 113 115 116 70 35 76  
## [217] 147 96 177 160 148 90 120 12 72 49 148 87 125 157 79 103 155 135  
## [235] 38 166 132 135 96 151 138 135 135 77 115 92 80 120 85 93 12 131  
## [253] 105 189 71 79 180 125 68 119 26 203 109 92 157 190 136 178 66 33  
## [271] 89 122 149 80 123 52 125 133 76 121 175 54 114 157 64 109 92 63  
## [289] 80 97 81 76 195 85 37 57 78 128 77 134 127 86 115 64 146 135  
## [307] 66 56 120 120 140 80 28 112 173 54 117 66 16 70 106 96 127 72  
## [325] 99 106 121 45 102 88 111 25 187 175 64 73 128 122 108 133 140 24  
## [343] 57 52 107 21 177 113 62 55 97 132 67 30 137 121 151 151 194 15  
## [361] 93 74 73 110 62 94 28 81 80 130 110 93 111 165 57 113 66 105  
## [379] 137 88 117 175 126 104 68 160 131 134 163 40 117 147 28 170 91 154  
## [397] 113 90 83 92 153 136 113 145 58 42 20 139 191 133 122 136 134 97  
## [415] 124 22 65 126 137 151 200 125 139 23 88 194 129 78 116 95 106 99  
## [433] 179 37 118 142 140 141 70 128 53 111 137 64 107 44 73 153 102 127  
## [451] 54 67 79 88 84 111 61 124 142 46 148 72 87 82 177 104 109 53  
## [469] 30 100 103 133 42 167 101 114 156 87 153 70 175 102 116 86 99 163  
## [487] 27 83 68 118 48 145 110 117 102 148 84 177 82 78 105 96 102 106  
## [505] 120 144 179 99 95 35 182 124 126 207 91 115 95 109 128 121 169 142  
## [523] 53 113 178 3 153 132 90 138 137 110 108 87 147 131 75 77 131 81  
## [541] 94 27 122 121 139 49 232 154 101 50 117 130 163 121 109 147 217 76  
## [559] 34 174 128 85 104 166 110 111 132 37 128 87 103 159 109 72 44 170  
## [577] 121 97 113 174 62 78 101 194 83 31 109 98 30 48 110 93 144 134  
## [595] 155 162 115 93 127 88 100 128 88 147 145 143 87 50 86 122 163 147  
## [613] 149 109 149 98 62 112 155 131 179 53 77 46 123 125 115 123 81 102  
## [631] 78 113 94 189 88 140 184 171 68 63 101 87 101 144 128 186 137 171  
## [649] 93 125 61 134 122 161 112 134 134 133 112 99 130 113 59 73 182 116  
## [667] 45 91 68 33 66 52 98 70 109 78 75 95 147 213 208 134 48 96  
## [685] 65 156 106 45 123 106 99 48 127 77 83 117 158 96 94 88 59 137  
## [703] 73 69 91 125 171 27 172 56 166 111 152 177 87 86 90 93 85 50  
## [721] 49 108 76 129 51 66 101 198 149 96 140 78 122 83 153 67 78 48  
## [739] 87 156 132 96 76 61 132 83 79 149 125 144 39 60 120 100 106 116  
## [757] 64 24 62 113 77 112 88 82 84 137 118 19 148 151 167 60 20 118  
## [775] 119 122 55 128 116 124 148 98 73 206 48 161 78 105 132 160 115 108  
## [793] 122 27 110 124 105 54 87 163 149 169 115 108 104 70 93 104 43 99  
## [811] 60 15 87 172 60 167 75 90 68 123 93 115 169 123 88 80 132 135  
## [829] 10 93 59 86 167 123 163 97 51 101 124 93 84 140 186 199 41 67  
## [847] 163 99 78 134 159 101 101 71 62 156 133 4 105 112 130 72 136 209  
## [865] 33 131 73 110 59 21 142 129 78 144 197 182 74 18 25 153 210 59  
## [883] 73 138 149 58 101 21 149 86 134 88 174 91 96 90 140 112 126 174  
## [901] 150 50 94 159 100 204 80 84 132 78 182 92 161 131 156 138 135 83  
## [919] 97 83 117 43 75 90 129 159 87 86 147 79 82 153 85 78 159 21  
## [937] 63 160 173 156 88 120 76 92 27 99 97 98 94 148 50 42 94 96  
## [955] 120 88 136 39 5 160 80 91 78 84 225 127 127 132 151 100 43 91  
## [973] 63 21 45 110 100 89 124 15 245 81 101 140 151 104 139 100 75 130  
## [991] 41 113 97 101 117 64 55 161 75 144 121 127 139 85 67 130 108 231  
## [1009] 97 103 94 158 101 53 104 52 109 174 74 110 124 46 3 129 160 125  
## [1027] 130 147 32 77 103 148 91 88 87 95 24 121 170 111 110 51 117 111  
## [1045] 94 99 64 103 154 91 99 45 96 29 196 71 74 42 100 89 69 69  
## [1063] 157 95 87 120 85 4 86 95 108 177 174 189 107 32 41 92 126 56  
## [1081] 103 96 46 107 118 195 145 79 96 81 63 121 144 36 89 167 169 19  
## [1099] 117 164 138 124 189 136 123 165 150 124 101 120 159 120 37 95 131 90  
## [1117] 109 88 105 106 108 142 34 101 16 133 52 80 87 146 113 67 118 111  
## [1135] 136 106 125 121 32 212 104 104 162 62 147 102 119 129 122 131 114 112  
## [1153] 134 51 34 90 151 76 84 84 58 89 92 65 136 130 108 117 98 149  
## [1171] 176 47 74 123 88 67 149 143 136 138 113 188 141 43 97 68 92 133  
## [1189] 111 133 189 110 95 74 13 74 131 118 138 149 126 97 115 135 103 82  
## [1207] 133 77 150 125 62 88 87 139 75 75 107 138 119 146 157 160 172 56  
## [1225] 110 22 76 51 75 140 100 76 64 95 241 143 149 108 60 92 96 194  
## [1243] 142 87 64 155 87 64 61 117 80 138 174 119 95 84 101 176 169 167  
## [1261] 50 110 70 160 159 151 38 151 68 136 105 50 124 72 128 157 108 111  
## [1279] 87 52 52 27 149 138 120 100 58 130 126 113 146 137 163 101 10 107  
## [1297] 131 116 115 137 82 129 108 130 104 116 87 50 154 85 116 125 93 129  
## [1315] 4 71 146 76 33 55 180 77 124 130 134 107 105 64 68 193 93 86  
## [1333] 104 121 120 72 55 63 108 181 119 163 107 127 94 97 85 92 97 178  
## [1351] 176 103 137 125 92 68 100 51 112 88 123 112 62 51 155 51 117 95  
## [1369] 101 180 168 215 115 161 74 83 88 72 74 127 36 27 160 111 110 70  
## [1387] 173 92 146 85 36 78 143 77 137 114 143 106 38 145 99 88 98 152  
## [1405] 106 109 99 76 143 48 160 148 35 52 92 112 48 110 152 144 87 106  
## [1423] 5 86 155 113 99 119 124 102 176 113 66 95 39 112 108 148 87 174  
## [1441] 151 179 160 117 101 85 127 151 55 96 105 146 161 105 115 94 215 95  
## [1459] 74 169 69 125 163 129 125 70 124 89 179 102 35 71 117 100 131 71  
## [1477] 122 50 161 161 39 192 90 35 103 105 134 61 117 126 82 119 57 83  
## [1495] 147 160 137 106 125 86 69 127 138 90 74 121 129 167 121 207 172 110  
## [1513] 224 72 123 94 99 164 138 87 91 135 121 36 147 87 111 123 113 139  
## [1531] 107 115 158 205 149 177 101 124 93 67 136 66 162 132 94 91 28 77  
## [1549] 132 65 32 119 127 85 200 63 106 97 93 103 90 130 93 104 100 75  
## [1567] 49 112 126 110 129 124 132 59 129 190 43 48 91 96 150 70 99 95  
## [1585] 142 34 112 125 62 99 108 65 75 71 106 76 75 140 71 148 44 216  
## [1603] 107 119 70 158 87 105 84 127 93 96 156 42 80 132 107 142 74 186  
## [1621] 141 158 78 53 84 118 106 160 101 129 150 84 79 95 45 72 78 90  
## [1639] 19 47 44 115 84 90 103 146 105 102 94 156 96 140 40 79 83 107  
## [1657] 114 159 74 122 75 76 69 91 104 120 116 196 64 121 129 73 120 107  
## [1675] 84 35 164 167 154 134 43 90 94 169 41 120 162 181 67 111 141 65  
## [1693] 72 154 69 55 219 112 113 114 150 168 148 136 129 114 86 117 148 84  
## [1711] 134 78 112 123 136 119 117 122 132 143 91 90 54 76 152 56 35 87  
## [1729] 153 167 74 99 115 95 87 135 193 36 95 145 90 111 182 89 77 128  
## [1747] 139 164 107 52 138 94 212 156 139 143 77 70 80 121 66 130 148 140  
## [1765] 27 111 100 111 119 91 69 173 102 133 168 130 86 108 97 81 74 145  
## [1783] 112 95 100 111 93 21 116 151 99 148 166 120 93 140 70 98 124 181  
## [1801] 71 155 125 158 50 117 74 192 24 108 134 29 63 214 124 85 122 122  
## [1819] 52 109 93 127 86 181 57 117 132 91 105 71 62 166 81 112 100 139  
## [1837] 127 71 123 122 177 117 155 59 78 180 78 13 148 82 89 105 51 163  
## [1855] 55 101 30 121 117 129 73 106 123 115 71 130 148 60 50 128 137 26  
## [1873] 77 127 132 103 119 103 147 10 138 152 83 92 141 183 118 16 127 157  
## [1891] 133 117 121 84 92 83 127 112 129 118 59 150 74 138 90 122 53 97  
## [1909] 40 96 150 66 53 126 101 196 80 163 66 73 165 139 166 117 106 130  
## [1927] 102 148 159 166 54 132 68 80 101 89 118 143 127 75 101 60 116 101  
## [1945] 85 183 142 169 96 44 141 90 55 111 87 78 125 33 130 80 126 185  
## [1963] 88 85 111 97 72 81 60 96 83 147 71 76 48 126 98 106 210 43  
## [1981] 77 130 60 156 91 112 74 116 130 139 93 162 46 89 31 79 123 42  
## [1999] 108 102 145 62 130 144 24 25 46 61 69 74 111 165 64 104 67 67  
## [2017] 107 130 152 27 70 46 135 82 107 208 128 127 64 126 202 46 76 135  
## [2035] 156 169 157 111 79 69 178 155 63 100 86 106 72 57 55 159 85 37  
## [2053] 165 85 125 37 137 149 172 160 102 87 126 123 75 52 45 132 74 134  
## [2071] 101 204 121 152 87 120 106 63 184 120 85 98 136 129 72 150 157 146  
## [2089] 89 178 80 75 100 105 95 62 50 145 113 69 68 67 127 63 129 91  
## [2107] 30 116 90 34 68 191 117 118 148 169 140 118 117 55 152 80 182 107  
## [2125] 42 150 69 102 122 76 94 28 115 131 105 135 112 59 79 161 13 2  
## [2143] 147 59 145 88 106 126 145 133 125 160 68 69 94 168 124 76 149 86  
## [2161] 23 128 119 107 144 102 106 90 190 94 80 61 87 54 129 155 85 114  
## [2179] 56 31 33 124 121 67 105 138 84 91 144 82 96 82 115 141 75 133  
## [2197] 119 94 145 112 155 161 190 42 38 155 105 88 73 192 44 166 73 125  
## [2215] 92 102 149 72 125 79 113 70 69 97 100 73 85 54 174 46 110 62  
## [2233] 114 65 58 102 62 185 141 135 81 133 80 101 102 68 25 165 115 11  
## [2251] 124 131 40 154 138 123 48 4 156 137 37 88 187 95 90 73 13 83  
## [2269] 145 113 168 86 128 99 122 124 106 82 117 7 108 133 177 143 175 73  
## [2287] 72 101 58 67 97 137 103 119 115 120 72 122 126 14 119 148 12 17  
## [2305] 68 109 130 159 100 115 130 143 141 133 59 85 151 107 214 160 159 92  
## [2323] 155 76 95 125 99 121 177 59 103 58 123 135 81 23 113 94 78 134  
## [2341] 135 95 97 64 98 140 172 55 109 146 75 112 109 122 88 169 26 158  
## [2359] 125 104 102 132 139 73 145 94 71 157 138 136 63 13 141 76 122 101  
## [2377] 98 166 93 93 140 16 123 127 169 100 46 121 127 112 170 62 165 77  
## [2395] 159 165 138 83 16 71 62 82 141 136 150 58 104 152 84 200 102 143  
## [2413] 82 63 108 137 216 41 90 90 76 60 157 46 136 167 137 137 141 81  
## [2431] 136 37 98 132 146 161 104 190 141 152 131 115 124 107 119 92 83 148  
## [2449] 123 94 91 148 189 138 118 62 195 169 115 162 111 16 171 157 48 88  
## [2467] 195 156 38 199 128 112 60 64 86 124 124 155 60 91 105 99 57 150  
## [2485] 109 48 59 95 103 107 115 171 122 124 170 127 67 101 48 103 150 173  
## [2503] 101 21 114 73 104 124 94 140 110 73 73 28 38 76 49 82 75 44  
## [2521] 130 105 88 109 104 48 138 84 120 56 108 71 113 209 122 109 60 148  
## [2539] 79 37 43 86 143 212 134 85 138 134 194 95 76 65 45 131 54 101  
## [2557] 10 65 74 136 95 121 3 145 162 120 227 73 178 82 130 113 134 73  
## [2575] 108 18 90 99 113 95 104 109 94 88 3 105 124 129 13 225 74 151  
## [2593] 103 193 85 243 82 164 106 87 172 165 103 114 143 120 99 54 143 88  
## [2611] 104 116 90 90 99 162 134 52 176 141 119 101 142 131 127 173 168 162  
## [2629] 39 41

#Analysising Outliers

boxplot\_data = data  
boxplot\_data$state=NULL  
boxplot\_data$area\_code= NULL  
boxplot\_data$international\_plan=NULL  
boxplot\_data$voice\_mail\_plan=NULL  
boxplot\_data$churn=NULL  
  
boxplot(boxplot\_data)



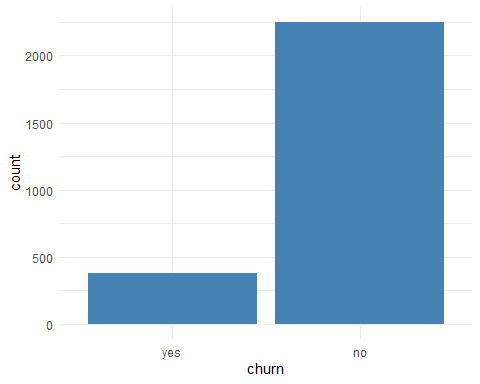
data$churn<- as.factor(data$churn)  
data$churn = factor(data$churn,levels(data$churn)[c(2,1)])  
summary(data$churn)

## yes no   
## 380 2250

str(data)

## tibble [2,630 × 20] (S3: tbl\_df/tbl/data.frame)  
## $ state : chr [1:2630] "NV" "HI" "DC" "OH" ...  
## $ account\_length : num [1:2630] 125 108 82 83 135 28 86 125 106 131 ...  
## $ area\_code : chr [1:2630] "area\_code\_510" "area\_code\_415" "area\_code\_415" "area\_code\_415" ...  
## $ international\_plan : chr [1:2630] "no" "no" "no" "no" ...  
## $ voice\_mail\_plan : chr [1:2630] "no" "no" "no" "no" ...  
## $ number\_vmail\_messages : num [1:2630] 0 0 0 0 0 0 0 0 9 0 ...  
## $ total\_day\_minutes : num [1:2630] 2013 292 300 337 202 ...  
## $ total\_day\_calls : num [1:2630] 99 99 109 120 81 87 115 108 100 95 ...  
## $ total\_day\_charge : num [1:2630] 28.7 49.6 51 57.4 34.3 ...  
## $ total\_eve\_minutes : num [1:2630] 1108 221 181 227 225 ...  
## $ total\_eve\_calls : num [1:2630] 107 93 100 116 114 92 112 90 109 109 ...  
## $ total\_eve\_charge : num [1:2630] 14.9 18.8 15.4 19.3 19.1 ...  
## $ total\_night\_minutes : num [1:2630] 243 229 270 154 204 ...  
## $ total\_night\_calls : num [1:2630] 92 110 73 114 82 112 95 114 86 129 ...  
## $ total\_night\_charge : num [1:2630] 10.95 10.31 12.15 6.93 9.2 ...  
## $ total\_intl\_minutes : num [1:2630] 10.9 14 11.7 15.8 10.3 10.1 9.8 13.3 8.2 15.5 ...  
## $ total\_intl\_calls : num [1:2630] 7 9 4 7 6 3 7 7 4 3 ...  
## $ total\_intl\_charge : num [1:2630] 2.94 3.78 3.16 4.27 2.78 2.73 2.65 3.59 2.21 4.19 ...  
## $ number\_customer\_service\_calls: num [1:2630] 0 2 0 0 1 3 2 1 3 1 ...  
## $ churn : Factor w/ 2 levels "yes","no": 2 1 1 1 2 2 2 2 2 1 ...  
## - attr(\*, "na.action")= 'omit' Named int [1:703] 4 6 10 12 13 14 23 32 33 39 ...  
## ..- attr(\*, "names")= chr [1:703] "4" "6" "10" "12" ...

ggplot(data) +  
 aes(x = churn) +  
 geom\_bar(fill = "#4682B4") +  
 theme\_minimal()



##Churn Train data partitioning (70%,30%)

partition<- createDataPartition(data$churn,p=0.7,list=FALSE)  
train<-data[partition,]  
validation<- data[-partition,]

# Accuracy for logistic regression

Model <- glm(churn ~ .,family=binomial(link="logit"),data=validation)  
summary(Model)

##   
## Call:  
## glm(formula = churn ~ ., family = binomial(link = "logit"), data = validation)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.08649 0.04634 0.21448 0.43557 2.52218   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.341e+01 2.371e+00 5.657 1.54e-08 \*\*\*  
## stateAL -9.527e-02 1.456e+00 -0.065 0.947844   
## stateAR -5.263e-01 1.545e+00 -0.341 0.733392   
## stateAZ -1.500e+00 1.543e+00 -0.972 0.330858   
## stateCA -2.201e-01 1.775e+00 -0.124 0.901338   
## stateCO -1.825e+00 1.459e+00 -1.251 0.211033   
## stateCT -7.428e-01 1.657e+00 -0.448 0.653851   
## stateDC -8.616e-01 1.693e+00 -0.509 0.610786   
## stateDE 1.558e+01 1.651e+03 0.009 0.992472   
## stateFL -8.500e-01 1.517e+00 -0.560 0.575217   
## stateGA 1.168e+00 1.864e+00 0.627 0.530836   
## stateHI -5.723e-01 1.517e+00 -0.377 0.705993   
## stateIA -2.125e+00 1.532e+00 -1.387 0.165292   
## stateID -9.701e-01 1.521e+00 -0.638 0.523653   
## stateIL 1.874e+00 1.761e+00 1.064 0.287162   
## stateIN 8.975e-02 1.689e+00 0.053 0.957611   
## stateKS -2.340e+00 1.448e+00 -1.616 0.106147   
## stateKY -3.762e-02 1.533e+00 -0.025 0.980427   
## stateLA -1.923e+00 1.462e+00 -1.315 0.188495   
## stateMA 1.507e+01 1.604e+03 0.009 0.992501   
## stateMD -7.425e-01 1.427e+00 -0.520 0.602888   
## stateME -7.141e-01 1.432e+00 -0.499 0.617908   
## stateMI -1.681e+00 1.616e+00 -1.040 0.298147   
## stateMN -1.981e-01 1.534e+00 -0.129 0.897270   
## stateMO -2.104e-01 1.780e+00 -0.118 0.905912   
## stateMS -1.902e+00 1.427e+00 -1.333 0.182553   
## stateMT -1.274e+00 1.517e+00 -0.839 0.401304   
## stateNC -3.941e-01 1.714e+00 -0.230 0.818201   
## stateND 3.217e-01 1.541e+00 0.209 0.834638   
## stateNE 1.589e+01 1.510e+03 0.011 0.991606   
## stateNH 1.484e+01 2.552e+03 0.006 0.995360   
## stateNJ -9.042e-01 1.515e+00 -0.597 0.550639   
## stateNM 1.589e+01 1.706e+03 0.009 0.992570   
## stateNV -1.878e+00 1.393e+00 -1.348 0.177531   
## stateNY -1.073e+00 1.468e+00 -0.731 0.464847   
## stateOH 1.264e+00 1.694e+00 0.746 0.455458   
## stateOK 7.477e-01 1.732e+00 0.432 0.665891   
## stateOR -1.403e+00 1.414e+00 -0.992 0.321042   
## statePA -1.157e+00 1.731e+00 -0.668 0.503924   
## stateRI 1.639e+00 1.908e+00 0.859 0.390343   
## stateSC -9.675e-01 1.666e+00 -0.581 0.561385   
## stateSD -1.364e+00 1.488e+00 -0.916 0.359529   
## stateTN 3.272e-01 1.667e+00 0.196 0.844426   
## stateTX -1.914e+00 1.470e+00 -1.302 0.193079   
## stateUT 8.364e-01 1.767e+00 0.473 0.635985   
## stateVA 1.575e-01 1.676e+00 0.094 0.925134   
## stateVT -1.206e-01 1.604e+00 -0.075 0.940080   
## stateWA -1.355e+00 1.551e+00 -0.874 0.382286   
## stateWI 1.635e+01 1.534e+03 0.011 0.991497   
## stateWV -1.793e+00 1.397e+00 -1.284 0.199297   
## stateWY -1.030e+00 1.388e+00 -0.742 0.457829   
## account\_length 2.595e-03 3.244e-03 0.800 0.423670   
## area\_codearea\_code\_415 1.521e-01 3.595e-01 0.423 0.672362   
## area\_codearea\_code\_510 4.287e-02 4.071e-01 0.105 0.916131   
## international\_planyes -2.717e+00 3.814e-01 -7.123 1.06e-12 \*\*\*  
## voice\_mail\_planyes 1.637e+00 8.624e-01 1.899 0.057627 .   
## number\_vmail\_messages -8.088e-03 2.806e-02 -0.288 0.773183   
## total\_day\_minutes 2.158e-04 4.894e-03 0.044 0.964831   
## total\_day\_calls -1.092e-02 7.099e-03 -1.539 0.123863   
## total\_day\_charge -1.039e-01 2.872e-02 -3.616 0.000299 \*\*\*  
## total\_eve\_minutes -6.140e-04 9.757e-03 -0.063 0.949820   
## total\_eve\_calls -2.447e-03 6.800e-03 -0.360 0.718938   
## total\_eve\_charge -1.538e-01 1.170e-01 -1.315 0.188659   
## total\_night\_minutes -9.327e-01 2.135e+00 -0.437 0.662220   
## total\_night\_calls -1.729e-03 6.738e-03 -0.257 0.797518   
## total\_night\_charge 2.065e+01 4.745e+01 0.435 0.663397   
## total\_intl\_minutes -1.896e+01 1.310e+01 -1.447 0.147948   
## total\_intl\_calls 3.937e-02 6.339e-02 0.621 0.534602   
## total\_intl\_charge 6.963e+01 4.853e+01 1.435 0.151340   
## number\_customer\_service\_calls -6.043e-01 9.799e-02 -6.166 6.99e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 651.75 on 788 degrees of freedom  
## Residual deviance: 412.46 on 719 degrees of freedom  
## AIC: 552.46  
##   
## Number of Fisher Scoring iterations: 17

predict\_validation<-predict(Model,newdata = validation,type='response')  
resultcheck<-ifelse(predict\_validation > 0.5, "no", "yes")  
  
  
resultcheck\_new<- as.factor(resultcheck)  
resultcheck\_new = factor(resultcheck\_new,levels(resultcheck\_new)[c(2,1)])  
  
New<-cbind(validation,resultcheck\_new)  
  
confusionMatrix(validation$churn,resultcheck\_new)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction yes no  
## yes 43 71  
## no 21 654  
##   
## Accuracy : 0.8834   
## 95% CI : (0.8589, 0.905)  
## No Information Rate : 0.9189   
## P-Value [Acc > NIR] : 0.9998   
##   
## Kappa : 0.4232   
##   
## Mcnemar's Test P-Value : 3.245e-07   
##   
## Sensitivity : 0.67188   
## Specificity : 0.90207   
## Pos Pred Value : 0.37719   
## Neg Pred Value : 0.96889   
## Prevalence : 0.08112   
## Detection Rate : 0.05450   
## Detection Prevalence : 0.14449   
## Balanced Accuracy : 0.78697   
##   
## 'Positive' Class : yes   
##

#Predicting churn value on Test data using the above model:

Predict\_data<-predict(Model,data=Customers\_To\_Predict,type="response")  
View(Predict\_data)

## Using AUC (Area under Curve)For model checking

roc(as.numeric(validation$churn), as.numeric(Predict\_data))

## Setting levels: control = 1, case = 2

## Setting direction: controls < cases

##   
## Call:  
## roc.default(response = as.numeric(validation$churn), predictor = as.numeric(Predict\_data))  
##   
## Data: as.numeric(Predict\_data) in 114 controls (as.numeric(validation$churn) 1) < 675 cases (as.numeric(validation$churn) 2).  
## Area under the curve: 0.8973

plot(roc(as.numeric(validation$churn), as.numeric(Predict\_data)))

## Setting levels: control = 1, case = 2  
## Setting direction: controls < cases

