EXPLORATORY DATA ANALYSIS USING R PROGRAMMING AND VARIOUS PACKAGES

Sunday BENJAMIN

January 22, 2021

library(plyr)  
library(dplyr)

library(tidyr)  
library(ggplot2)  
library(lubridate)

LOADING DATA SETS

hcustomerdata= read.csv("./ml\_case\_training\_data.csv")  
pricing\_data= read.csv("./ml\_case\_training\_hist\_data.csv")  
churn\_data = read.csv("./ml\_case\_training\_output.csv")

EXPLORATORY DATA ANALYSIS OF CUSTOMER DATA SET  
head(hcustomerdata, 2L)

## id activity\_new  
## 1 48ada52261e7cf58715202705a0451c9 esoiiifxdlbkcsluxmfuacbdckommixw  
## 2 24011ae4ebbe3035111d65fa7c15bc57   
## campaign\_disc\_ele channel\_sales cons\_12m cons\_gas\_12m  
## 1 NA lmkebamcaaclubfxadlmueccxoimlema 309275 0  
## 2 NA foosdfpfkusacimwkcsosbicdxkicaua 0 54946  
## cons\_last\_month date\_activ date\_end date\_first\_activ date\_modif\_prod  
## 1 10025 2012-11-07 2016-11-06 2012-11-07  
## 2 0 2013-06-15 2016-06-15   
## date\_renewal forecast\_base\_bill\_ele forecast\_base\_bill\_year forecast\_bill\_12m  
## 1 2015-11-09 NA NA NA  
## 2 2015-06-23 NA NA NA  
## forecast\_cons forecast\_cons\_12m forecast\_cons\_year forecast\_discount\_energy  
## 1 NA 26520.3 10025 0  
## 2 NA 0.0 0 0  
## forecast\_meter\_rent\_12m forecast\_price\_energy\_p1 forecast\_price\_energy\_p2  
## 1 359.29 0.095919 0.088347  
## 2 1.78 0.114481 0.098142  
## forecast\_price\_pow\_p1 has\_gas imp\_cons margin\_gross\_pow\_ele  
## 1 58.99595 f 831.8 -41.76  
## 2 40.60670 t 0.0 25.44  
## margin\_net\_pow\_ele nb\_prod\_act net\_margin num\_years\_antig  
## 1 -41.76 1 1732.36 3  
## 2 25.44 2 678.99 3  
## origin\_up pow\_max  
## 1 ldkssxwpmemidmecebumciepifcamkci 180.000  
## 2 lxidpiddsbxsbosboudacockeimpuepw 43.648

tail(hcustomerdata,2L)

## id activity\_new campaign\_disc\_ele  
## 16095 1cf20fd6206d7678d5bcafd28c53b4db NA  
## 16096 563dde550fd624d7352f3de77c0cdfcd NA  
## channel\_sales cons\_12m cons\_gas\_12m cons\_last\_month  
## 16095 foosdfpfkusacimwkcsosbicdxkicaua 131 0 0  
## 16096 8730 0 0  
## date\_activ date\_end date\_first\_activ date\_modif\_prod date\_renewal  
## 16095 2012-08-30 2016-08-30 2012-08-30 2015-08-31  
## 16096 2009-12-18 2016-12-17 2009-12-18 2015-12-21  
## forecast\_base\_bill\_ele forecast\_base\_bill\_year forecast\_bill\_12m  
## 16095 NA NA NA  
## 16096 NA NA NA  
## forecast\_cons forecast\_cons\_12m forecast\_cons\_year  
## 16095 NA 19.34 0  
## 16096 NA 762.41 0  
## forecast\_discount\_energy forecast\_meter\_rent\_12m forecast\_price\_energy\_p1  
## 16095 0 7.18 0.145711  
## 16096 0 1.07 0.167086  
## forecast\_price\_energy\_p2 forecast\_price\_pow\_p1 has\_gas imp\_cons  
## 16095 0.000000 44.31138 f 0  
## 16096 0.088454 45.31138 f 0  
## margin\_gross\_pow\_ele margin\_net\_pow\_ele nb\_prod\_act net\_margin  
## 16095 13.08 13.08 1 0.96  
## 16096 11.84 11.84 1 96.34  
## num\_years\_antig origin\_up pow\_max  
## 16095 3 lxidpiddsbxsbosboudacockeimpuepw 11.000  
## 16096 6 ldkssxwpmemidmecebumciepifcamkci 10.392

EXPLORATORY DATA ANALYSIS OF PRICING DATA SET  
head(pricing\_data,2L)

## id price\_date price\_p1\_var price\_p2\_var  
## 1 038af19179925da21a25619c5a24b745 01-01-15 0.151367 0  
## 2 038af19179925da21a25619c5a24b745 01-02-15 0.151367 0  
## price\_p3\_var price\_p1\_fix price\_p2\_fix price\_p3\_fix  
## 1 0 44.26693 0 0  
## 2 0 44.26693 0 0

tail(pricing\_data,2L)

## id price\_date price\_p1\_var price\_p2\_var  
## 193001 16f51cdc2baa19af0b940ee1b3dd17d5 01-11-15 0.119916 0.102232  
## 193002 16f51cdc2baa19af0b940ee1b3dd17d5 01-12-15 0.119916 0.102232  
## price\_p3\_var price\_p1\_fix price\_p2\_fix price\_p3\_fix  
## 193001 0.076257 40.72888 24.43733 16.29155  
## 193002 0.076257 40.72888 24.43733 16.29155

EXPLORATORY DATA ANALYSIS OF CHURN DATA SET  
head(churn\_data,2L)

## id churn  
## 1 48ada52261e7cf58715202705a0451c9 0  
## 2 24011ae4ebbe3035111d65fa7c15bc57 1

tail(churn\_data,2L)

## id churn  
## 16095 1cf20fd6206d7678d5bcafd28c53b4db 0  
## 16096 563dde550fd624d7352f3de77c0cdfcd 0

COMBINING HCUSTOMER DATA SET WITH CHURN DATA SET  
train = merge(hcustomerdata, churn\_data, all.x = T)

head(train, 2L)

## id activity\_new campaign\_disc\_ele  
## 1 0002203ffbb812588b632b9e628cc38d NA  
## 2 0004351ebdd665e6ee664792efc4fd13 NA  
## channel\_sales cons\_12m cons\_gas\_12m cons\_last\_month  
## 1 foosdfpfkusacimwkcsosbicdxkicaua 22034 0 3084  
## 2 4060 0 0  
## date\_activ date\_end date\_first\_activ date\_modif\_prod date\_renewal  
## 1 2010-01-19 2016-02-21 2010-01-19 2015-02-25  
## 2 2009-08-06 2016-06-21 2013-06-21 2015-06-23  
## forecast\_base\_bill\_ele forecast\_base\_bill\_year forecast\_bill\_12m  
## 1 NA NA NA  
## 2 NA NA NA  
## forecast\_cons forecast\_cons\_12m forecast\_cons\_year forecast\_discount\_energy  
## 1 NA 729.06 425 0  
## 2 NA 597.77 0 0  
## forecast\_meter\_rent\_12m forecast\_price\_energy\_p1 forecast\_price\_energy\_p2  
## 1 138.95 0.116900 0.100015  
## 2 6.84 0.142065 0.000000  
## forecast\_price\_pow\_p1 has\_gas imp\_cons margin\_gross\_pow\_ele  
## 1 40.60670 f 40.78 43.08  
## 2 44.31138 f 0.00 24.42  
## margin\_net\_pow\_ele nb\_prod\_act net\_margin num\_years\_antig  
## 1 43.08 1 81.42 6  
## 2 24.42 1 61.58 6  
## origin\_up pow\_max churn  
## 1 kamkkxfxxuwbdslkwifmmcsiusiuosws 17.25 0  
## 2 kamkkxfxxuwbdslkwifmmcsiusiuosws 13.20 0

tail(train, 2L)

## id activity\_new campaign\_disc\_ele  
## 16095 fffe4f5646aa39c7f97f95ae2679ce64 NA  
## 16096 ffff7fa066f1fb305ae285bb03bf325a NA  
## channel\_sales cons\_12m cons\_gas\_12m cons\_last\_month  
## 16095 32066 2916 4879  
## 16096 foosdfpfkusacimwkcsosbicdxkicaua 50806 0 5491  
## date\_activ date\_end date\_first\_activ date\_modif\_prod date\_renewal  
## 16095 2011-09-07 2016-09-06 2011-09-07 2015-09-07  
## 16096 2012-06-20 2016-06-20 2013-11-05 2015-06-23  
## forecast\_base\_bill\_ele forecast\_base\_bill\_year forecast\_bill\_12m  
## 16095 NA NA NA  
## 16096 NA NA NA  
## forecast\_cons forecast\_cons\_12m forecast\_cons\_year  
## 16095 NA 3313.13 4879  
## 16096 NA 1038.70 1057  
## forecast\_discount\_energy forecast\_meter\_rent\_12m forecast\_price\_energy\_p1  
## 16095 0 130.31 0.115174  
## 16096 0 131.02 0.116910  
## forecast\_price\_energy\_p2 forecast\_price\_pow\_p1 has\_gas imp\_cons  
## 16095 0.098837 40.6067 t 487.59  
## 16096 0.100572 40.6067 f 103.02  
## margin\_gross\_pow\_ele margin\_net\_pow\_ele nb\_prod\_act net\_margin  
## 16095 19.68 19.68 3 361.4  
## 16096 23.72 23.72 1 132.2  
## num\_years\_antig origin\_up pow\_max churn  
## 16095 4 lxidpiddsbxsbosboudacockeimpuepw 31.5 0  
## 16096 4 lxidpiddsbxsbosboudacockeimpuepw 19.0 0

## DATA TYPES

glimpse(train)

## Rows: 16,096  
## Columns: 33  
## $ id <chr> "0002203ffbb812588b632b9e628cc38d", "00043...  
## $ activity\_new <chr> "", "", "fskfsbkdioupwobbsaoospkxaafmwobl"...  
## $ campaign\_disc\_ele <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ channel\_sales <chr> "foosdfpfkusacimwkcsosbicdxkicaua", "", "u...  
## $ cons\_12m <int> 22034, 4060, 7440, 4199490, 11272, 104657,...  
## $ cons\_gas\_12m <int> 0, 0, 0, 728810, 0, 0, 0, 0, 57630, 0, 0, ...  
## $ cons\_last\_month <int> 3084, 0, 1062, 456462, 0, 6760, 19394, 550...  
## $ date\_activ <chr> "2010-01-19", "2009-08-06", "2013-02-25", ...  
## $ date\_end <chr> "2016-02-21", "2016-06-21", "2016-05-05", ...  
## $ date\_first\_activ <chr> "", "", "", "", "", "", "2013-02-22", "", ...  
## $ date\_modif\_prod <chr> "2010-01-19", "2013-06-21", "2015-05-05", ...  
## $ date\_renewal <chr> "2015-02-25", "2015-06-23", "2015-02-26", ...  
## $ forecast\_base\_bill\_ele <dbl> NA, NA, NA, NA, NA, NA, 302.04, NA, NA, NA...  
## $ forecast\_base\_bill\_year <dbl> NA, NA, NA, NA, NA, NA, 302.04, NA, NA, NA...  
## $ forecast\_bill\_12m <dbl> NA, NA, NA, NA, NA, NA, 4553.78, NA, NA, N...  
## $ forecast\_cons <dbl> NA, NA, NA, NA, NA, NA, 195.20, NA, NA, NA...  
## $ forecast\_cons\_12m <dbl> 729.06, 597.77, 1311.16, 11776.27, 1671.41...  
## $ forecast\_cons\_year <int> 425, 0, 1062, 17393, 0, 6760, 1760, 5501, ...  
## $ forecast\_discount\_energy <dbl> 0, 0, 30, 0, 0, 0, 0, 0, 0, 0, 0, 30, 0, 0...  
## $ forecast\_meter\_rent\_12m <dbl> 138.95, 6.84, 18.37, 132.11, 18.27, 393.44...  
## $ forecast\_price\_energy\_p1 <dbl> 0.116900, 0.142065, 0.199230, 0.110083, 0....  
## $ forecast\_price\_energy\_p2 <dbl> 0.100015, 0.000000, 0.000000, 0.093746, 0....  
## $ forecast\_price\_pow\_p1 <dbl> 40.60670, 44.31138, 45.80688, 40.60670, 44...  
## $ has\_gas <chr> "f", "f", "f", "t", "f", "f", "f", "f", "t...  
## $ imp\_cons <dbl> 40.78, 0.00, 213.76, 1533.07, 0.00, 642.89...  
## $ margin\_gross\_pow\_ele <dbl> 43.08, 24.42, 38.58, -2.80, 29.76, -4.41, ...  
## $ margin\_net\_pow\_ele <dbl> 43.08, 24.42, 38.58, -2.80, 29.76, -4.41, ...  
## $ nb\_prod\_act <int> 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, ...  
## $ net\_margin <dbl> 81.42, 61.58, 81.61, 897.08, 157.99, 700.7...  
## $ num\_years\_antig <int> 6, 6, 3, 6, 6, 4, 3, 3, 12, 3, 6, 5, 7, 4,...  
## $ origin\_up <chr> "kamkkxfxxuwbdslkwifmmcsiusiuosws", "kamkk...  
## $ pow\_max <dbl> 17.250, 13.200, 13.856, 33.000, 13.200, 70...  
## $ churn <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, ...

glimpse(pricing\_data)

## Rows: 193,002  
## Columns: 8  
## $ id <chr> "038af19179925da21a25619c5a24b745", "038af19179925da21...  
## $ price\_date <chr> "01-01-15", "01-02-15", "01-03-15", "01-04-15", "01-05...  
## $ price\_p1\_var <dbl> 0.151367, 0.151367, 0.151367, 0.149626, 0.149626, 0.14...  
## $ price\_p2\_var <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.00...  
## $ price\_p3\_var <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.00...  
## $ price\_p1\_fix <dbl> 44.26693, 44.26693, 44.26693, 44.26693, 44.26693, 44.2...  
## $ price\_p2\_fix <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, ...  
## $ price\_p3\_fix <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, ...

## DATA FRAME STATS

apply(train %>% select(5:7,13:23,25:30,32:33),2, mean)

## cons\_12m cons\_gas\_12m cons\_last\_month   
## 1.948044e+05 3.191164e+04 1.946154e+04   
## forecast\_base\_bill\_ele forecast\_base\_bill\_year forecast\_bill\_12m   
## NA NA NA   
## forecast\_cons forecast\_cons\_12m forecast\_cons\_year   
## NA 2.370556e+03 1.907347e+03   
## forecast\_discount\_energy forecast\_meter\_rent\_12m forecast\_price\_energy\_p1   
## NA 7.030994e+01 NA   
## forecast\_price\_energy\_p2 forecast\_price\_pow\_p1 imp\_cons   
## NA NA 1.961234e+02   
## margin\_gross\_pow\_ele margin\_net\_pow\_ele nb\_prod\_act   
## NA NA 1.347788e+00   
## net\_margin num\_years\_antig pow\_max   
## NA 5.030629e+00 NA   
## churn   
## 9.909294e-02

apply(train %>% select(5:7,13:23,25:30,32:33),2, sd)

## cons\_12m cons\_gas\_12m cons\_last\_month   
## 6.795151e+05 1.775885e+05 8.235676e+04   
## forecast\_base\_bill\_ele forecast\_base\_bill\_year forecast\_bill\_12m   
## NA NA NA   
## forecast\_cons forecast\_cons\_12m forecast\_cons\_year   
## NA 4.035086e+03 5.257365e+03   
## forecast\_discount\_energy forecast\_meter\_rent\_12m forecast\_price\_energy\_p1   
## NA 7.902325e+01 NA   
## forecast\_price\_energy\_p2 forecast\_price\_pow\_p1 imp\_cons   
## NA NA 4.943670e+02   
## margin\_gross\_pow\_ele margin\_net\_pow\_ele nb\_prod\_act   
## NA NA 1.459808e+00   
## net\_margin num\_years\_antig pow\_max   
## NA 1.676101e+00 NA   
## churn   
## 2.987960e-01

apply(na.omit(train %>% select(5:7,13:23,25:30,32:33)),2,min) ## na.omit removes NA's

## cons\_12m cons\_gas\_12m cons\_last\_month   
## -17957.0000 -3037.0000 -12035.0000   
## forecast\_base\_bill\_ele forecast\_base\_bill\_year forecast\_bill\_12m   
## -364.9400 -364.9400 -2503.4800   
## forecast\_cons forecast\_cons\_12m forecast\_cons\_year   
## 0.0000 -2882.5300 0.0000   
## forecast\_discount\_energy forecast\_meter\_rent\_12m forecast\_price\_energy\_p1   
## 0.0000 -114.9100 0.0006   
## forecast\_price\_energy\_p2 forecast\_price\_pow\_p1 imp\_cons   
## 0.0000 0.0000 0.0000   
## margin\_gross\_pow\_ele margin\_net\_pow\_ele nb\_prod\_act   
## -254.5200 -293.4900 1.0000   
## net\_margin num\_years\_antig pow\_max   
## -3711.4000 1.0000 3.4640   
## churn   
## 0.0000

apply(train %>% select(5:7,13:23,25:30,32:33),2,max)

## cons\_12m cons\_gas\_12m cons\_last\_month   
## 16097108.00 4188440.00 4538720.00   
## forecast\_base\_bill\_ele forecast\_base\_bill\_year forecast\_bill\_12m   
## NA NA NA   
## forecast\_cons forecast\_cons\_12m forecast\_cons\_year   
## NA 103801.93 175375.00   
## forecast\_discount\_energy forecast\_meter\_rent\_12m forecast\_price\_energy\_p1   
## NA 2411.69 NA   
## forecast\_price\_energy\_p2 forecast\_price\_pow\_p1 imp\_cons   
## NA NA 15042.79   
## margin\_gross\_pow\_ele margin\_net\_pow\_ele nb\_prod\_act   
## NA NA 32.00   
## net\_margin num\_years\_antig pow\_max   
## NA 16.00 NA   
## churn   
## 1.00

apply(train %>% select(5:7,13:23,25:30,32:33),2, quantile, c(0.5,.75,1), na.rm=T)

## cons\_12m cons\_gas\_12m cons\_last\_month forecast\_base\_bill\_ele  
## 50% 15332.5 0 901 162.955  
## 75% 50221.5 0 4127 396.185  
## 100% 16097108.0 4188440 4538720 12566.080  
## forecast\_base\_bill\_year forecast\_bill\_12m forecast\_cons forecast\_cons\_12m  
## 50% 162.955 2187.230 42.2150 1179.160  
## 75% 396.185 4246.555 228.1175 2692.078  
## 100% 12566.080 81122.630 9682.8900 103801.930  
## forecast\_cons\_year forecast\_discount\_energy forecast\_meter\_rent\_12m  
## 50% 378.00 0 19.44  
## 75% 1994.25 0 131.47  
## 100% 175375.00 50 2411.69  
## forecast\_price\_energy\_p1 forecast\_price\_energy\_p2 forecast\_price\_pow\_p1  
## 50% 0.142881 0.086163 44.31138  
## 75% 0.146348 0.098837 44.31138  
## 100% 0.273963 0.195975 59.44471  
## imp\_cons margin\_gross\_pow\_ele margin\_net\_pow\_ele nb\_prod\_act net\_margin  
## 50% 44.465 21.09 20.97 1 119.68  
## 75% 218.090 29.64 29.64 1 275.81  
## 100% 15042.790 374.64 374.64 32 24570.65  
## num\_years\_antig pow\_max churn  
## 50% 5 13.856 0  
## 75% 6 19.800 0  
## 100% 16 500.000 1

## For pricing data

apply(na.omit(pricing\_data %>% select(-1,-2)),2,mean)

## price\_p1\_var price\_p2\_var price\_p3\_var price\_p1\_fix price\_p2\_fix price\_p3\_fix   
## 0.14099147 0.05441161 0.03071226 43.32554620 10.69820076 6.45543648

apply(na.omit(pricing\_data %>% select(-1,-2)),2,sd)

## price\_p1\_var price\_p2\_var price\_p3\_var price\_p1\_fix price\_p2\_fix price\_p3\_fix   
## 0.02511744 0.05003308 0.03633520 5.43795225 12.85604627 7.78227857

apply(na.omit(pricing\_data %>% select(-1,-2)),2,min)

## price\_p1\_var price\_p2\_var price\_p3\_var price\_p1\_fix price\_p2\_fix price\_p3\_fix   
## 0.0000000 0.0000000 0.0000000 -0.1777788 -0.0977520 -0.0651720

apply(na.omit(pricing\_data %>% select(-1,-2)),2,max)

## price\_p1\_var price\_p2\_var price\_p3\_var price\_p1\_fix price\_p2\_fix price\_p3\_fix   
## 0.280700 0.229788 0.114102 59.444710 36.490692 17.458221

apply(pricing\_data %>% select(-1,-2),2,quantile, c(0.5,0.75,1.00), na.rm=T) # na.omit was not used

## price\_p1\_var price\_p2\_var price\_p3\_var price\_p1\_fix price\_p2\_fix  
## 50% 0.146033 0.085483 0.000000 44.26693 0.00000  
## 75% 0.151635 0.101780 0.072558 44.44471 24.33958  
## 100% 0.280700 0.229788 0.114102 59.44471 36.49069  
## price\_p3\_fix  
## 50% 0.00000  
## 75% 16.22639  
## 100% 17.45822

apply(na.omit(pricing\_data %>% select(-1,-2)),2,quantile, c(0.5,0.75,1.00)) # na.omit was used

## price\_p1\_var price\_p2\_var price\_p3\_var price\_p1\_fix price\_p2\_fix  
## 50% 0.146033 0.085483 0.000000 44.26693 0.00000  
## 75% 0.151635 0.101780 0.072558 44.44471 24.33958  
## 100% 0.280700 0.229788 0.114102 59.44471 36.49069  
## price\_p3\_fix  
## 50% 0.00000  
## 75% 16.22639  
## 100% 17.45822

## Missing Values in train data set

apply(train, 2, function(col)sum(is.na(col))/length(col)\*100)

## id activity\_new campaign\_disc\_ele   
## 0.00000000 0.00000000 100.00000000   
## channel\_sales cons\_12m cons\_gas\_12m   
## 0.00000000 0.00000000 0.00000000   
## cons\_last\_month date\_activ date\_end   
## 0.00000000 0.00000000 0.00000000   
## date\_first\_activ date\_modif\_prod date\_renewal   
## 0.00000000 0.00000000 0.00000000   
## forecast\_base\_bill\_ele forecast\_base\_bill\_year forecast\_bill\_12m   
## 78.20576541 78.20576541 78.20576541   
## forecast\_cons forecast\_cons\_12m forecast\_cons\_year   
## 78.20576541 0.00000000 0.00000000   
## forecast\_discount\_energy forecast\_meter\_rent\_12m forecast\_price\_energy\_p1   
## 0.78280318 0.00000000 0.78280318   
## forecast\_price\_energy\_p2 forecast\_price\_pow\_p1 has\_gas   
## 0.78280318 0.78280318 0.00000000   
## imp\_cons margin\_gross\_pow\_ele margin\_net\_pow\_ele   
## 0.00000000 0.08076541 0.08076541   
## nb\_prod\_act net\_margin num\_years\_antig   
## 0.00000000 0.09319085 0.00000000   
## origin\_up pow\_max churn   
## 0.00000000 0.01863817 0.00000000

## Don't use (i.e drop) any Column that has more than 75% Missing values

## Missing Values for pricing data

apply(pricing\_data,2, function(col) sum(is.na(col))/length(col)\*100)

## id price\_date price\_p1\_var price\_p2\_var price\_p3\_var price\_p1\_fix   
## 0.0000000 0.0000000 0.7041378 0.7041378 0.7041378 0.7041378   
## price\_p2\_fix price\_p3\_fix   
## 0.7041378 0.7041378

## Deep Visualization

ftable(xtabs(churn~., data = churn\_data))

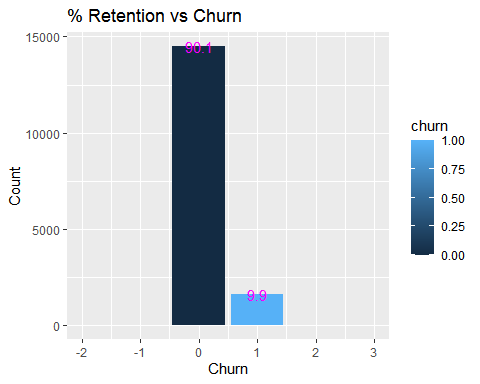
## 0 1  
##   
## 14501 1595

table\_plot=churn\_data %>%group\_by(churn) %>% summarise(n=n())

## `summarise()` ungrouping output (override with `.groups` argument)

table\_plot %>%  
 ggplot(aes(churn,n, fill=churn))+  
 geom\_col(position="dodge")+  
 labs(title="% Retention vs Churn",  
 x="Churn", y= "Count")+  
 geom\_text(aes(label = round(n/16096\*100, 1)),  
 position = position\_dodge(4),  
 color="magenta",vjust = 0.5,hjust = 0.5)

## Warning: position\_dodge requires non-overlapping x intervals



## SME Activity

activity = train %>%   
 group\_by(activity\_new,churn,id) %>%   
 select(activity\_new, churn,id) %>%   
 summarise(n=n()) %>%   
 summarise(n=n()) %>%   
 spread("churn","n")

activity= activity[-1, ] ## removal of 1st rows  
activity[is.na(activity)]=0 ## substitues NA's with Zero  
class(activity)

## [1] "grouped\_df" "tbl\_df" "tbl" "data.frame"

activity=as.data.frame(activity)  
class(activity)

## [1] "data.frame"

colnames(activity) = c("activity\_new", "retention","churn")

## 

## activity dataset with % Churn and % Retention

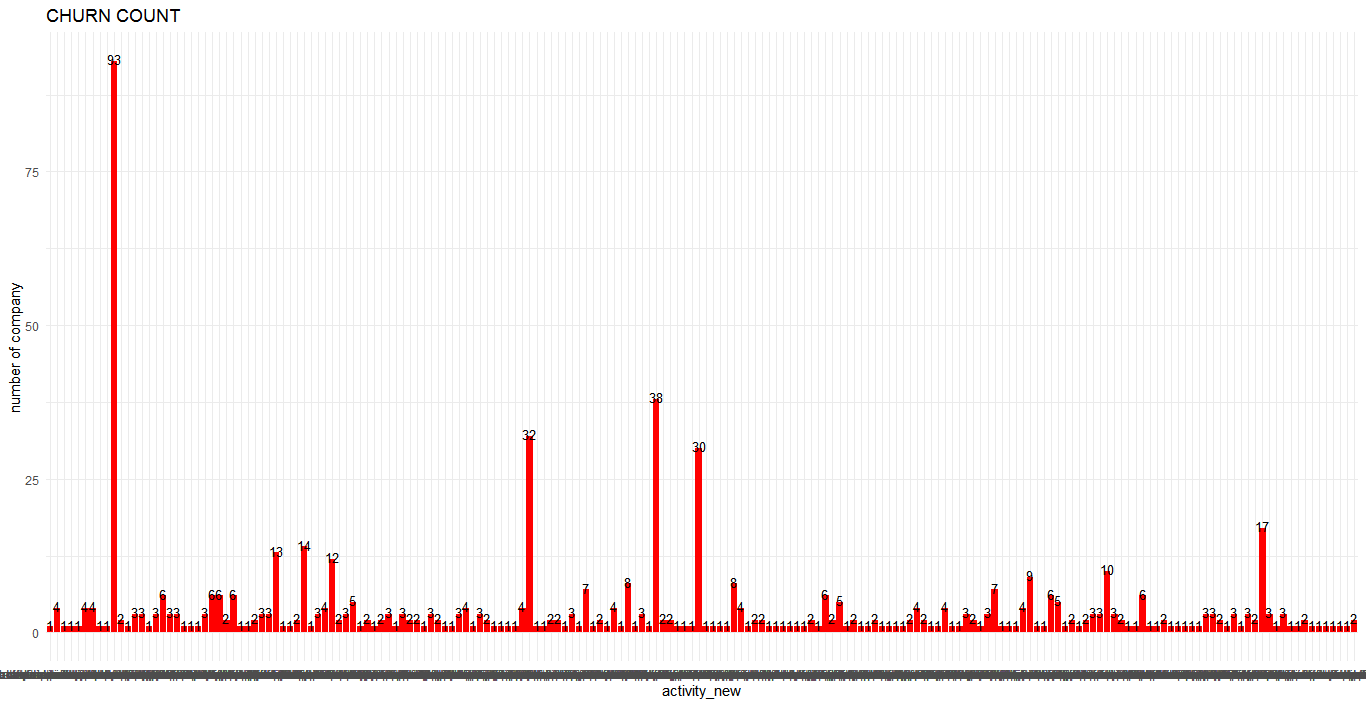
head(activity %>% mutate(Percentage\_churn = churn/rowSums(activity[ ,-1])\*100,   
 Percentage\_retention = 100-Percentage\_churn,   
 total\_no\_of\_company= rowSums(activity[ ,-1])) %>%   
 select(activity\_new,retention,Percentage\_retention,  
 churn,Percentage\_churn,total\_no\_of\_company), 2L)

## activity\_new retention Percentage\_retention churn  
## 1 aacewucldmklslcffeckexipaemmsdfk 1 100 0  
## 2 aamfdbbldmixubpkwmdacapsfexcksdo 3 100 0  
## Percentage\_churn total\_no\_of\_company  
## 1 0 1  
## 2 0 3

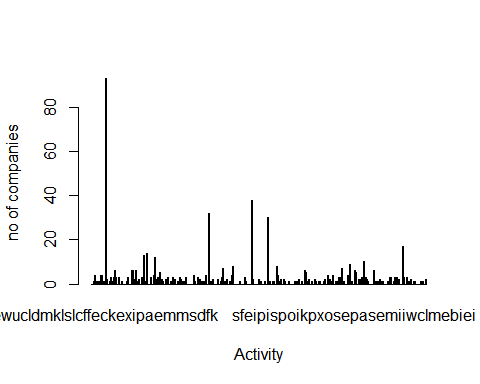
## 

## Visualization for Churn

activity %>%   
 filter(churn>=1) %>%   
 arrange(churn) %>%   
 ggplot(aes(x=activity\_new, y = churn)) +  
 geom\_bar(stat="identity", fill="red")+  
 labs(title="CHURN COUNT",x="activity\_new", y= "number of company")+  
 geom\_text(aes(label=churn), vjust=0.3, size=3.5)+  
 theme\_minimal()

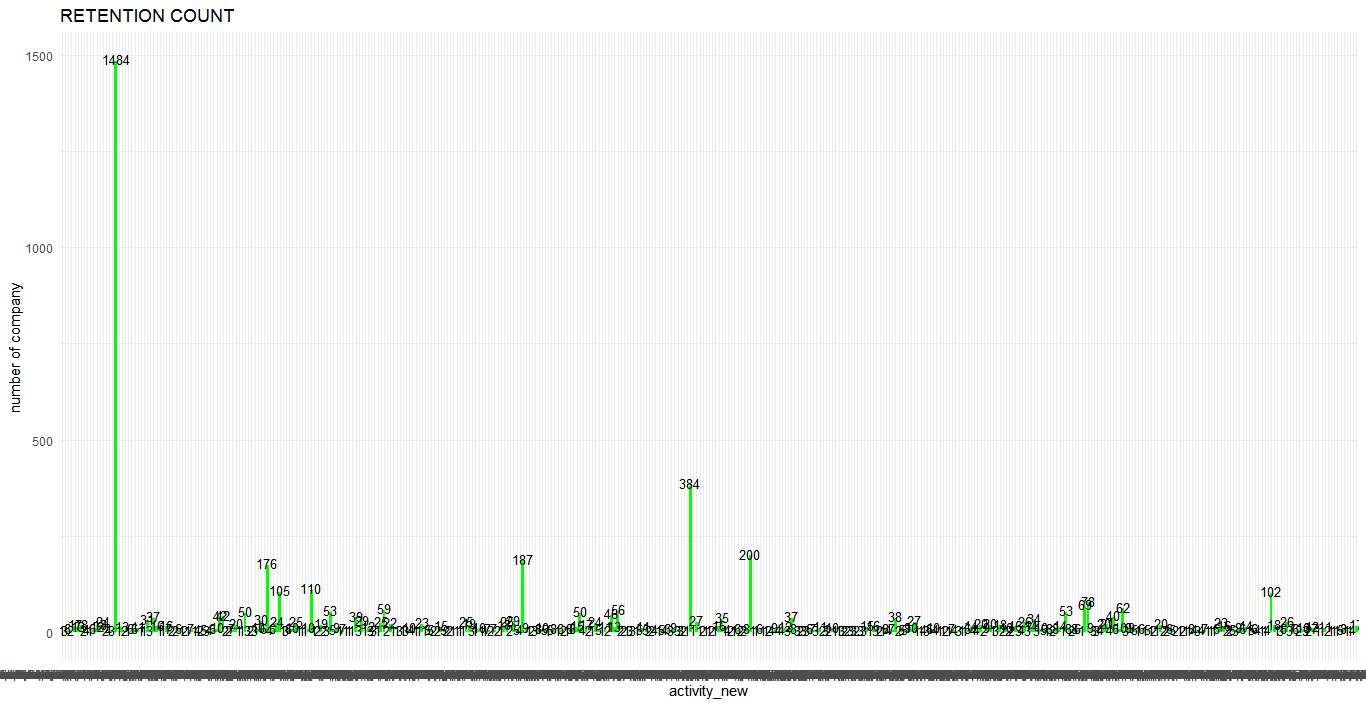


## OR

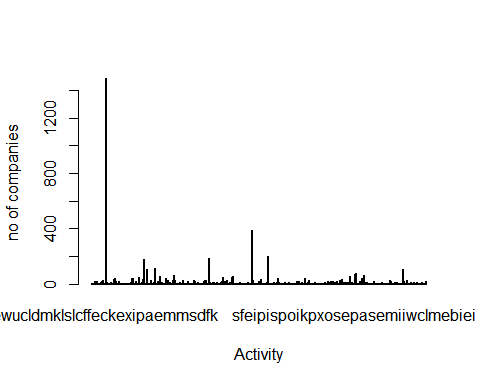
barplot(activity$churn, names.arg = activity$activity\_new,  
 xlab = "Activity", ylab = "no of companies")

## Visualization for Retention

activity %>%   
 filter(retention>=1) %>%   
 arrange(desc(retention)) %>%   
 ggplot(aes(x=activity\_new, y=retention)) +  
 geom\_bar(stat="identity", fill="green")+  
 labs(title="RETENTION COUNT",x="activity\_new", y= "number of company")+  
 geom\_text(aes(label=retention), vjust=0.3, size=3.5)+  
 theme\_minimal()



OR

barplot(activity$retention, names.arg = activity$activity\_new,  
 xlab = "Activity", ylab = "no of companies")

## 

## SALES CHANNEL

sales = train %>%   
 select(channel\_sales, churn,id) %>%   
 group\_by(channel\_sales,churn,id) %>%   
 summarise(n=n()) %>%   
 summarise(n=n()) %>%   
 spread(churn,n)

## `summarise()` regrouping output by 'channel\_sales', 'churn' (override with `.groups` argument)

## `summarise()` regrouping output by 'channel\_sales' (override with `.groups` argument)

sales = sales[-1, ]  
sales[is.na(sales)]=0  
sales = as.data.frame(sales)  
  
colnames(sales) = c("channel\_sales","retention", "churn")

## 

## Channel\_Sales dataset with % Churn and % Retention is REQUIRED

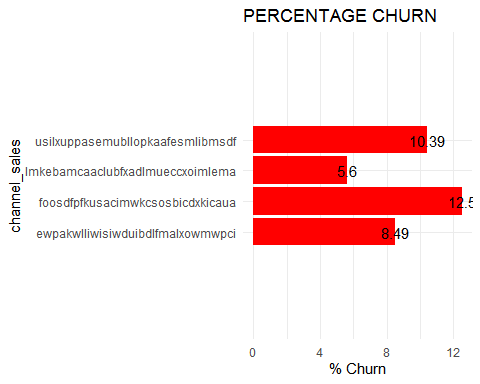
sales\_2= sales %>% mutate(Percent\_churn = round(churn/rowSums(sales[ ,-1])\*100, digits = 2),  
 Percent\_retained = 100-Percent\_churn,  
 total\_no\_of\_coy = rowSums(sales[ ,-1])) %>%   
 select(channel\_sales,retention,Percent\_retained,  
 churn,Percent\_churn,total\_no\_of\_coy)

## Bar Plots

## Visualization for Churn

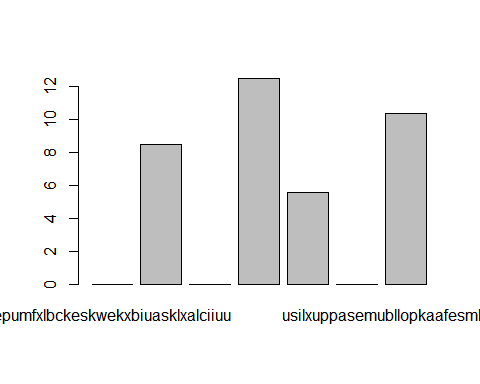
sales\_2 %>%   
 filter(churn>=1) %>%   
 arrange(Percent\_churn,channel\_sales) %>%   
 ggplot(aes(x=channel\_sales, y = Percent\_churn)) +  
 geom\_bar(stat="identity", fill="red")+  
 labs(title="PERCENTAGE CHURN",x="channel\_sales", y= "% Churn")+  
 geom\_text(aes(label=Percent\_churn), position = position\_dodge(7),  
 vjust=0.5,hjust = 0.5)+  
 theme\_minimal()+  
 coord\_flip()

## Warning: position\_dodge requires non-overlapping x intervals



## OR

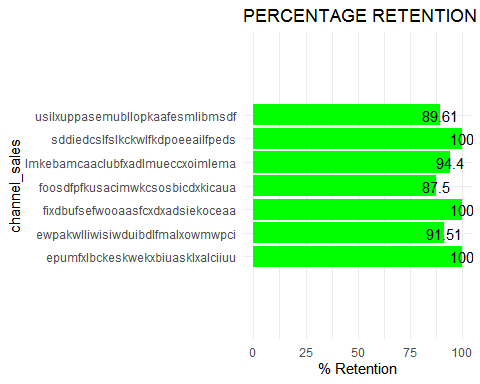
barplot(sales\_2$Percent\_churn, names.arg = sales\_2$channel\_sales)



## Visualization for Retention

sales\_2 %>%   
 filter(retention>=1) %>%   
 arrange(Percent\_retained,channel\_sales) %>%   
 ggplot(aes(x=channel\_sales, y = Percent\_retained)) +  
 geom\_bar(stat="identity", fill="GREEN")+  
 labs(title="PERCENTAGE RETENTION",x="channel\_sales", y= "% Retention")+  
 geom\_text(aes(label=Percent\_retained), position = position\_dodge(7),  
 vjust=0.5,hjust = 0.5)+  
 theme\_minimal()+  
 coord\_flip()

## Warning: position\_dodge requires non-overlapping x intervals



## 

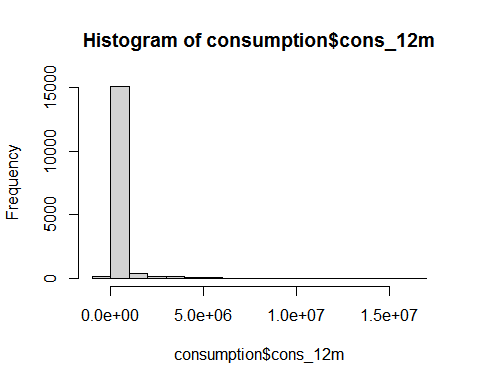
## Consumption Distribution

consumption= train %>%   
 select(id,cons\_12m,cons\_gas\_12m,cons\_last\_month,imp\_cons,has\_gas,churn)

## Histogram for cons\_12m

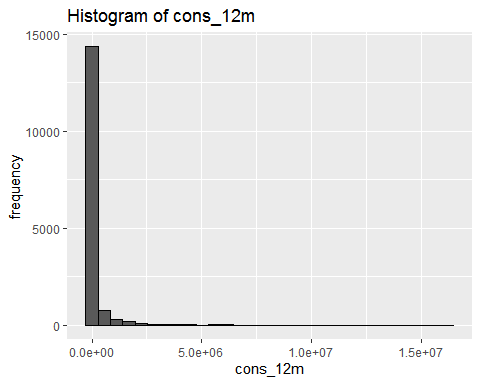
## For Total frequency distribution (Histogram) i.e churn + retention

hist(consumption$cons\_12m)



## OR

qplot(consumption$cons\_12m, geom = "histogram",  
 colour=I("black"),   
 xlab = "cons\_12m",   
 ylab = "frequency",   
 main = "Histogram of cons\_12m")

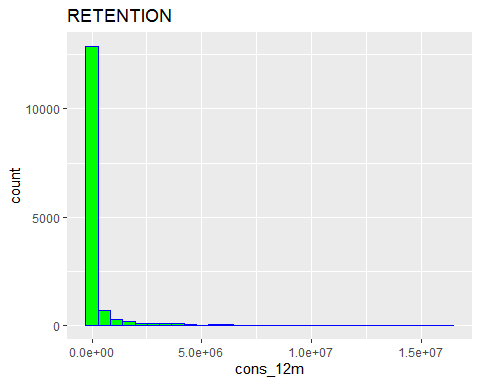


## To calculate churn and retention separately

## Histogram for cons\_12m (RETENTION)

consumption %>% filter(churn==0) %>%   
 ggplot(aes(cons\_12m)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")

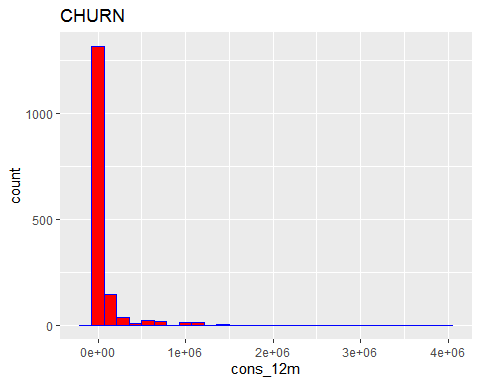
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Histogram for cons\_12m (CHURN)

consumption %>% filter(churn==1) %>%   
 ggplot(aes(cons\_12m)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

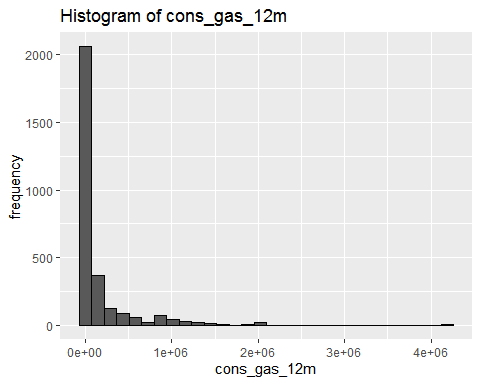


## Histogram for has\_gas=T and cons\_gas\_12\_m

## For Total (churn + retention) Frequency/Count Distribution

t\_gas=consumption %>% filter(has\_gas=="t") %>% select(cons\_gas\_12m,has\_gas)  
  
qplot(t\_gas$cons\_gas\_12m, geom="histogram",  
 colour=I("black"),  
 xlab = "cons\_gas\_12m",  
 ylab = "frequency",  
 main = "Histogram of cons\_gas\_12m")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

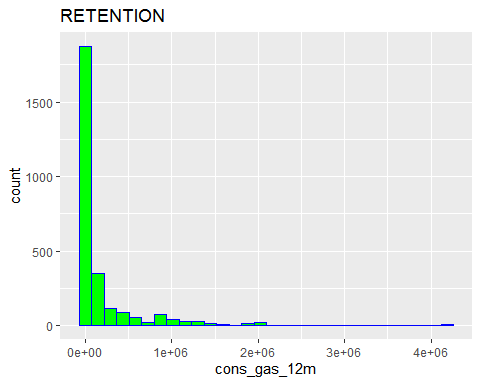


## To calculate churn and retention separately

## Histogram for cons\_gas\_12m (RETENTION)

consumption %>% filter(has\_gas=="t", churn==0) %>%   
 ggplot(aes(cons\_gas\_12m)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")

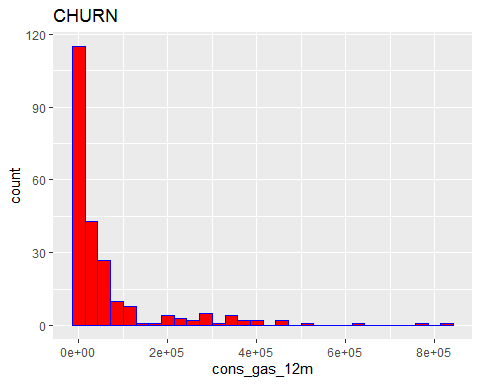
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Histogram for cons\_gas\_12m (CHURN)

consumption %>% filter(has\_gas=="t",churn==1) %>%   
 ggplot(aes(cons\_gas\_12m)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



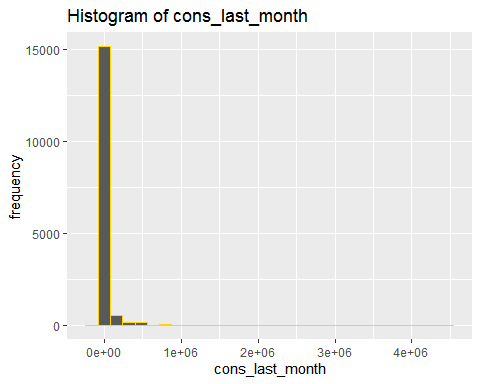
## 

## Histogram for cons\_last\_month

## For Total (churn + retention) Frequency/Count Distribution

qplot(consumption$cons\_last\_month, geom = "histogram",  
 color = I("GOLD"),  
 xlab = "cons\_last\_month",  
 ylab = "frequency",  
 main = "Histogram of cons\_last\_month")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

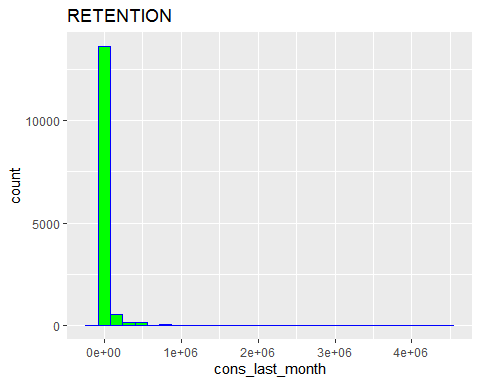


## To calculate churn and retention separately

## Histogram for cons\_last\_month (RETENTION)

consumption %>% filter(churn==0) %>%   
 ggplot(aes(cons\_last\_month)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")

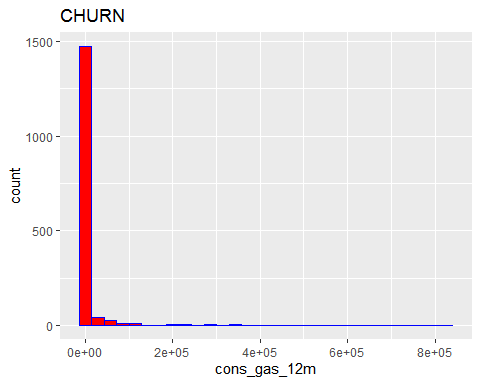
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Histogram for cons\_gas\_12m (CHURN)

consumption %>% filter(churn==1) %>%   
 ggplot(aes(cons\_gas\_12m)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



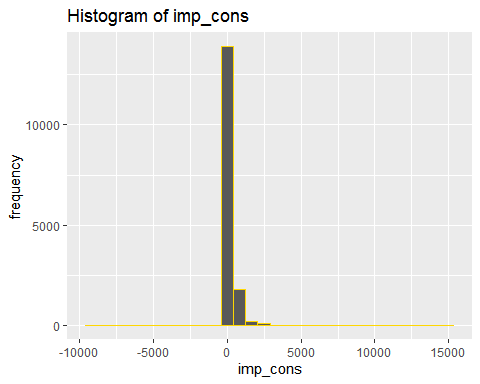
## 

## Histogram for imp\_cons

## For Total (churn + retention) Frequency/Count Distribution

qplot(consumption$imp\_cons, geom = "histogram", # cld input binwidth=40 to get more insights  
 color = I("GOLD"),  
 xlab = "imp\_cons",  
 ylab = "frequency",  
 main = "Histogram of imp\_cons")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



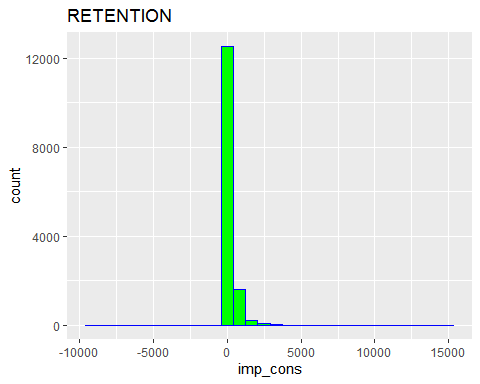
## 

## To calculate churn and retention separately

## Histogram for imp\_cons (RETENTION)

consumption %>% filter(churn==0) %>%   
 ggplot(aes(imp\_cons)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")

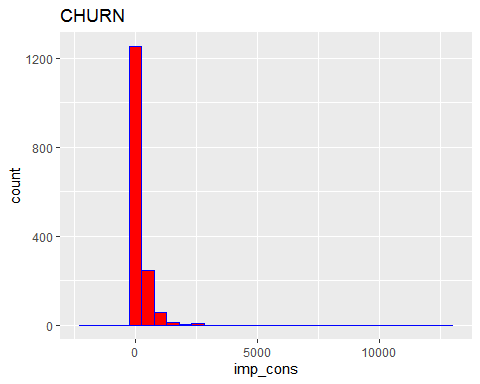
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Histogram for imp\_cons (CHURN)

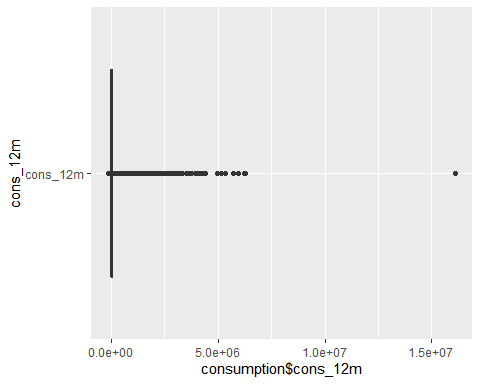
consumption %>% filter(churn==1) %>%   
 ggplot(aes(imp\_cons)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

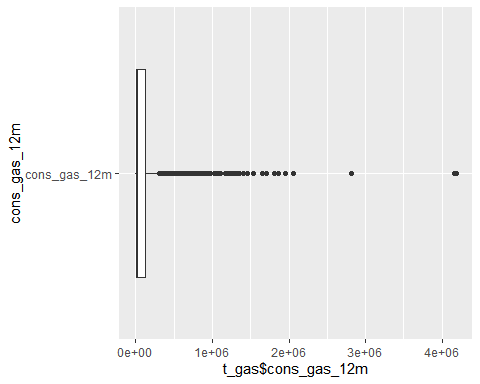


## Box Plots

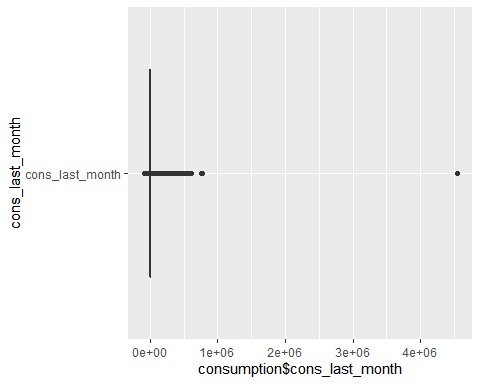
qplot("cons\_12m", consumption$cons\_12m, geom = "boxplot") + coord\_flip()



qplot("cons\_gas\_12m", t\_gas$cons\_gas\_12m, geom = "boxplot") + coord\_flip()



qplot("cons\_last\_month", consumption$cons\_last\_month, geom = "boxplot") + coord\_flip()



qplot("imp\_cons", consumption$imp\_cons,geom = "boxplot") + coord\_flip()



## Dates

dates = train %>%   
 select(id, date\_activ, date\_end, date\_modif\_prod, date\_renewal, churn)

glimpse(dates)

## Rows: 16,096  
## Columns: 6  
## $ id <chr> "0002203ffbb812588b632b9e628cc38d", "0004351ebdd665...  
## $ date\_activ <chr> "2010-01-19", "2009-08-06", "2013-02-25", "2010-06-...  
## $ date\_end <chr> "2016-02-21", "2016-06-21", "2016-05-05", "2016-06-...  
## $ date\_modif\_prod <chr> "2010-01-19", "2013-06-21", "2015-05-05", "2010-06-...  
## $ date\_renewal <chr> "2015-02-25", "2015-06-23", "2015-02-26", "2015-06-...  
## $ churn <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, ...

summary(dates)

## id date\_activ date\_end date\_modif\_prod   
## Length:16096 Length:16096 Length:16096 Length:16096   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## date\_renewal churn   
## Length:16096 Min. :0.00000   
## Class :character 1st Qu.:0.00000   
## Mode :character Median :0.00000   
## Mean :0.09909   
## 3rd Qu.:0.00000   
## Max. :1.00000

dates$date\_activ = as.Date(dates$date\_activ)  
 dates$date\_activ\_Year\_Month = format(dates$date\_activ, "%Y-%m")

dates$date\_end = as.Date(dates$date\_end)  
 dates$date\_end\_Year\_Month = format(dates$date\_end, "%Y-%m")

dates$date\_modif\_prod = as.Date(dates$date\_modif\_prod)  
 dates$date\_modif\_prod\_Year\_Month = format(dates$date\_modif\_prod, "%Y-%m")

dates$date\_renewal = as.Date(dates$date\_renewal)  
 dates$date\_renewal\_Year\_Month = format(dates$date\_renewal,"%Y-%m")

glimpse(dates)

## Rows: 16,096  
## Columns: 10  
## $ id <chr> "0002203ffbb812588b632b9e628cc38d", "000...  
## $ date\_activ <date> 2010-01-19, 2009-08-06, 2013-02-25, 201...  
## $ date\_end <date> 2016-02-21, 2016-06-21, 2016-05-05, 201...  
## $ date\_modif\_prod <date> 2010-01-19, 2013-06-21, 2015-05-05, 201...  
## $ date\_renewal <date> 2015-02-25, 2015-06-23, 2015-02-26, 201...  
## $ churn <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0...  
## $ date\_activ\_Year\_Month <chr> "2010-01", "2009-08", "2013-02", "2010-0...  
## $ date\_end\_Year\_Month <chr> "2016-02", "2016-06", "2016-05", "2016-0...  
## $ date\_modif\_prod\_Year\_Month <chr> "2010-01", "2013-06", "2015-05", "2010-0...  
## $ date\_renewal\_Year\_Month <chr> "2015-02", "2015-06", "2015-02", "2015-0...

summary(dates)

## id date\_activ date\_end   
## Length:16096 Min. :2000-07-25 Min. :2006-08-26   
## Class :character 1st Qu.:2010-01-12 1st Qu.:2016-04-28   
## Mode :character Median :2011-03-04 Median :2016-07-30   
## Mean :2011-01-17 Mean :2016-07-27   
## 3rd Qu.:2012-04-26 3rd Qu.:2016-10-31   
## Max. :2014-09-01 Max. :2017-06-13   
## NA's :2   
## date\_modif\_prod date\_renewal churn   
## Min. :2000-07-25 Min. :2013-06-26 Min. :0.00000   
## 1st Qu.:2010-08-10 1st Qu.:2015-04-19 1st Qu.:0.00000   
## Median :2013-05-01 Median :2015-07-24 Median :0.00000   
## Mean :2012-12-14 Mean :2015-07-20 Mean :0.09909   
## 3rd Qu.:2015-05-24 3rd Qu.:2015-10-30 3rd Qu.:0.00000   
## Max. :2016-01-29 Max. :2016-01-28 Max. :1.00000   
## NA's :157 NA's :40   
## date\_activ\_Year\_Month date\_end\_Year\_Month date\_modif\_prod\_Year\_Month  
## Length:16096 Length:16096 Length:16096   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
## date\_renewal\_Year\_Month  
## Length:16096   
## Class :character   
## Mode :character   
##   
##   
##   
##

## 

## Plotting Dates

colSums(is.na(dates))

## id date\_activ   
## 0 0   
## date\_end date\_modif\_prod   
## 2 157   
## date\_renewal churn   
## 40 0   
## date\_activ\_Year\_Month date\_end\_Year\_Month   
## 0 2   
## date\_modif\_prod\_Year\_Month date\_renewal\_Year\_Month   
## 157 40

d1 = dates %>%   
 group\_by(date\_activ\_Year\_Month,churn,id) %>%  
 select(date\_activ\_Year\_Month,churn,id) %>%   
 summarise(n=n()) %>%   
 summarise(n=n()) %>%   
 spread("churn", "n")

d1[is.na(d1)]=0

class(d1)

## [1] "grouped\_df" "tbl\_df" "tbl" "data.frame"

d1 = as.data.frame(d1) ## RATE LIMITING STEP; VERY IMPORTANT  
   
colnames(d1) = c("date\_activ\_Year\_Month","retention","churn")

## Percentage Calculations

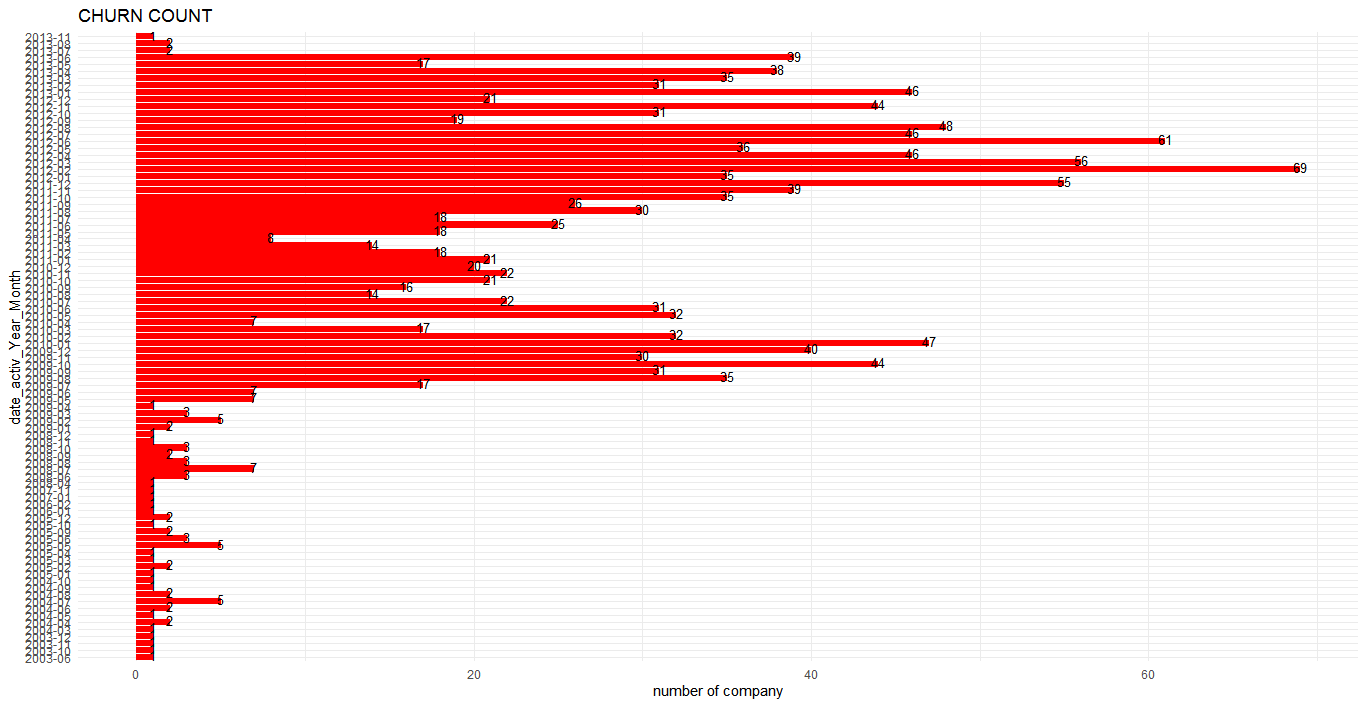
head(d1 %>% mutate(percentage\_churn = churn/rowSums(d1[ ,-1])\*100, ## d1[ ,-1] is to allow for computation  
 percentage\_retention = 100-percentage\_churn,  
 Total\_no\_company = rowSums(d1[,-1])) %>%   
 select(date\_activ\_Year\_Month,retention,percentage\_retention,  
 churn,percentage\_churn,Total\_no\_company),4L)

## date\_activ\_Year\_Month retention percentage\_retention churn percentage\_churn  
## 1 2000-07 1 100.00000 0 0.00000  
## 2 2001-02 1 100.00000 0 0.00000  
## 3 2003-05 1 100.00000 0 0.00000  
## 4 2003-06 2 66.66667 1 33.33333  
## Total\_no\_company  
## 1 1  
## 2 1  
## 3 1  
## 4 3

## 

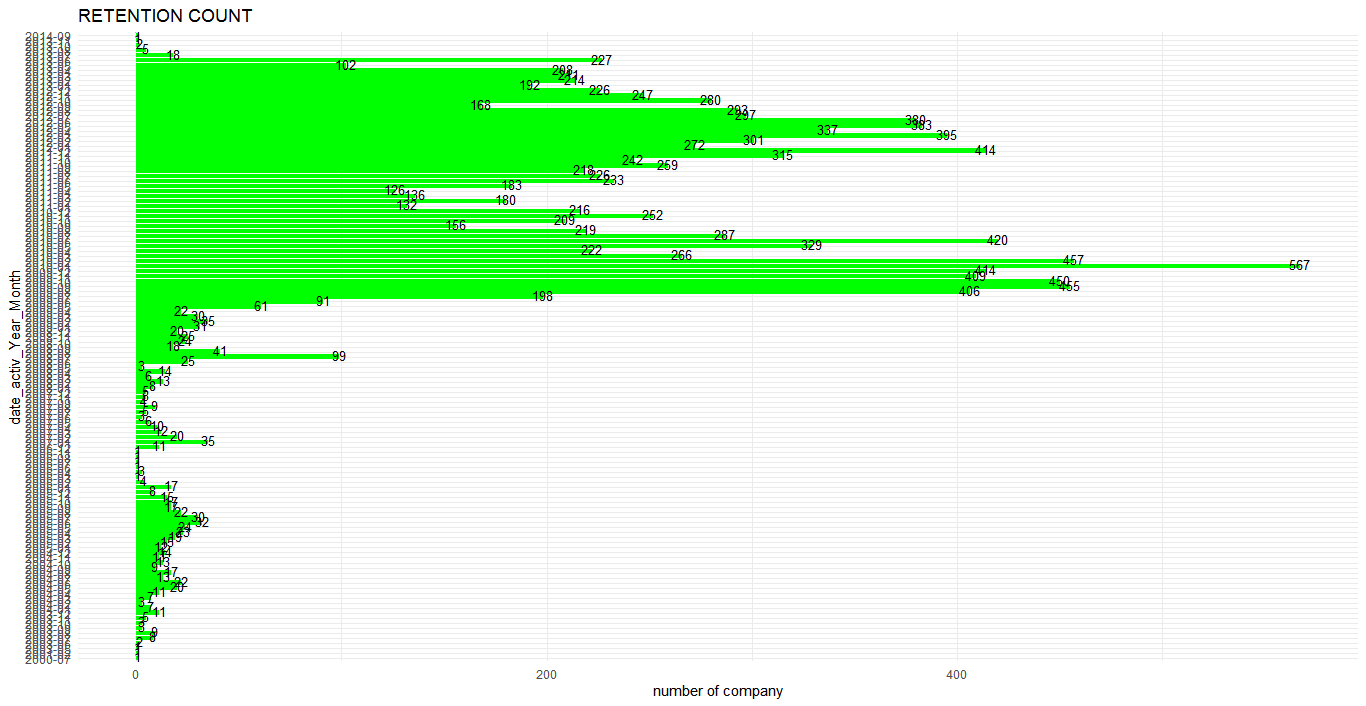
## Visualization for Churn

d1 %>% filter(churn>=1) %>% ## This line of code would take out the zero's  
 ggplot(aes(x=date\_activ\_Year\_Month, y = churn)) +  
 geom\_bar(stat="identity", fill="red")+  
 labs(title="CHURN COUNT",x="date\_activ\_Year\_Month", y= "number of company")+  
 geom\_text(aes(label=churn), vjust=0.3, size=3.5)+  
 theme\_minimal()+ coord\_flip()



## Visualization for Retention

d1 %>% filter(retention>=1) %>%   
 ggplot(aes(x=date\_activ\_Year\_Month, y=retention)) +  
 geom\_bar(stat="identity", fill="green")+  
 labs(title="RETENTION COUNT",x="date\_activ\_Year\_Month", y= "number of company")+  
 geom\_text(aes(label=retention), vjust=0.3, size=3.5)+  
 theme\_minimal()+coord\_flip()



## date\_end

d2 = dates %>%   
 group\_by(date\_end\_Year\_Month,churn,id) %>%  
 select(date\_end\_Year\_Month,churn,id) %>%   
 summarise(n=n()) %>% summarise(n=n()) %>%   
 spread("churn", "n")

## `summarise()` regrouping output by 'date\_end\_Year\_Month', 'churn' (override with `.groups` argument)

## `summarise()` regrouping output by 'date\_end\_Year\_Month' (override with `.groups` argument)

d2 = d2[-17, ]  
d2[is.na(d2)]=0  
class(d2)

## [1] "grouped\_df" "tbl\_df" "tbl" "data.frame"

d2 = as.data.frame(d2) ## RATE LIMITING STEP; VERY IMPORTANT  
colnames(d2) = c("date\_end\_Year\_Month","retention","churn")

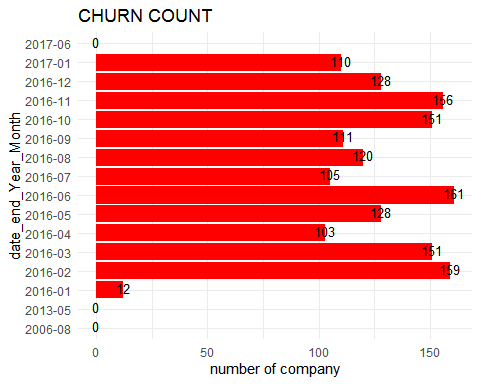
## Percentages Calculation

head(d2 %>% mutate(percentage\_churn = churn/rowSums(d2[ ,-1])\*100, ## d3[ ,-1] is to allow for computation  
 percentage\_retention = 100-percentage\_churn,  
 Total\_no\_company = rowSums(d2[,-1])) %>%   
 select(date\_end\_Year\_Month,retention,percentage\_retention,  
 churn,percentage\_churn,Total\_no\_company), 4L)

## date\_end\_Year\_Month retention percentage\_retention churn percentage\_churn  
## 1 2006-08 1 100.00000 0 0.00000  
## 2 2013-05 1 100.00000 0 0.00000  
## 3 2016-01 97 88.99083 12 11.00917  
## 4 2016-02 1300 89.10212 159 10.89788  
## Total\_no\_company  
## 1 1  
## 2 1  
## 3 109  
## 4 1459

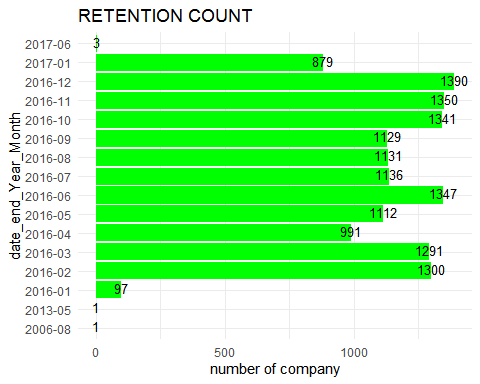
## Visualization for Churn

d2 %>%  
 ggplot(aes(x=date\_end\_Year\_Month, y = churn)) +  
 geom\_bar(stat="identity", fill="red")+  
 labs(title="CHURN COUNT",x="date\_end\_Year\_Month", y= "number of company")+  
 geom\_text(aes(label=churn), vjust=0.3, size=3.5)+  
 theme\_minimal()+coord\_flip()



## Visualization for Retention

d2 %>%   
 ggplot(aes(x=date\_end\_Year\_Month, y=retention)) +  
 geom\_bar(stat="identity", fill="green")+  
 labs(title="RETENTION COUNT",x="date\_end\_Year\_Month", y= "number of company")+  
 geom\_text(aes(label=retention), vjust=0.3, size=3.5)+  
 theme\_minimal()+coord\_flip()



## date\_modif\_prod

d3 = dates %>%   
 group\_by(date\_modif\_prod\_Year\_Month,churn,id) %>%  
 select(date\_modif\_prod\_Year\_Month,churn,id) %>%   
 summarise(n=n()) %>% summarise(n=n()) %>%   
 spread("churn", "n")

## `summarise()` regrouping output by 'date\_modif\_prod\_Year\_Month', 'churn' (override with `.groups` argument)

## `summarise()` regrouping output by 'date\_modif\_prod\_Year\_Month' (override with `.groups` argument)

d3 = d3[-149, ]  
d3[is.na(d3)]=0  
class(d3)

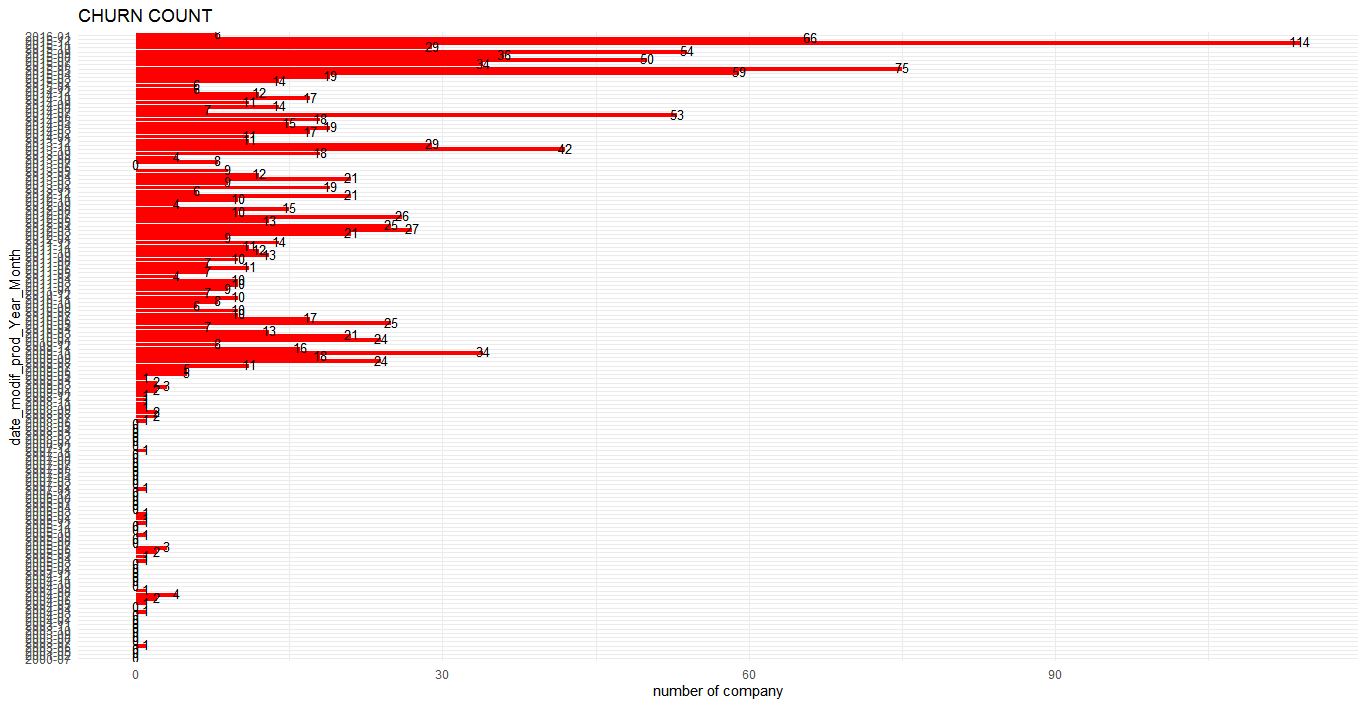
## [1] "grouped\_df" "tbl\_df" "tbl" "data.frame"

d3 = as.data.frame(d3) ## RATE LIMITING STEP; VERY IMPORTANT  
colnames(d3) = c("date\_modif\_prod\_Year\_Month","retention","churn")

head(d3 %>% mutate(percentage\_churn = churn/rowSums(d3[ ,-1])\*100, ## d3[ ,-1] is to allow for computation  
 percentage\_retention=100-percentage\_churn,  
 Total\_no\_company= rowSums(d3[,-1])) %>%   
 select(date\_modif\_prod\_Year\_Month,retention,percentage\_retention,  
 churn,percentage\_churn,Total\_no\_company))

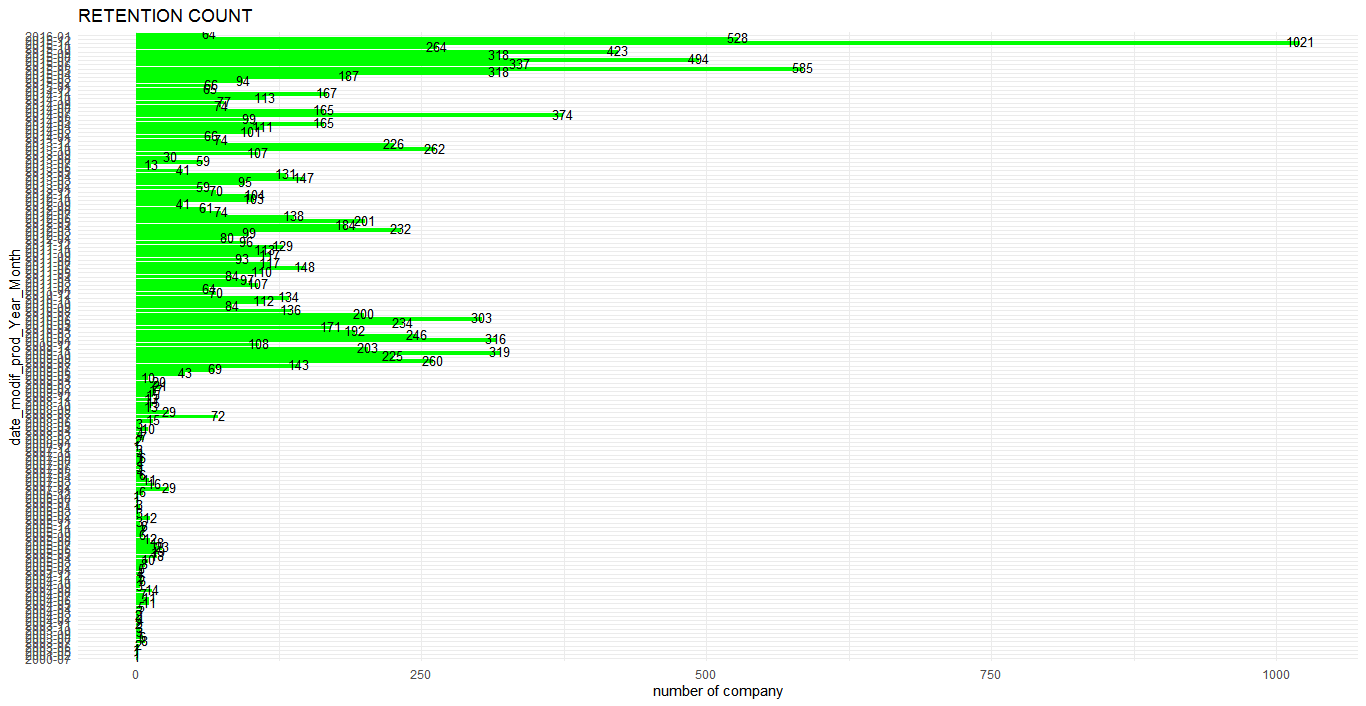
## date\_modif\_prod\_Year\_Month retention percentage\_retention churn  
## 1 2000-07 1 100.00000 0  
## 2 2001-02 1 100.00000 0  
## 3 2003-05 1 100.00000 0  
## 4 2003-06 2 66.66667 1  
## 5 2003-07 8 100.00000 0  
## 6 2003-08 6 100.00000 0  
## percentage\_churn Total\_no\_company  
## 1 0.00000 1  
## 2 0.00000 1  
## 3 0.00000 1  
## 4 33.33333 3  
## 5 0.00000 8  
## 6 0.00000 6

## Visualization for Churn

d3 %>%  
 ggplot(aes(x=date\_modif\_prod\_Year\_Month, y = churn)) +  
 geom\_bar(stat="identity", fill="red")+  
 labs(title="CHURN COUNT",x="date\_modif\_prod\_Year\_Month", y= "number of company")+  
 geom\_text(aes(label=churn), vjust=0.3, size=3.5)+  
 theme\_minimal()+coord\_flip()

## Visualization for Retention

d3 %>%   
 ggplot(aes(x=date\_modif\_prod\_Year\_Month, y=retention)) +  
 geom\_bar(stat="identity", fill="green")+  
 labs(title="RETENTION COUNT",x="date\_modif\_prod\_Year\_Month", y= "number of company")+  
 geom\_text(aes(label=retention), vjust=0.3, size=3.5)+  
 theme\_minimal()+coord\_flip()



## date\_renewal

d4 = dates %>%   
 group\_by(date\_renewal\_Year\_Month,churn,id) %>%  
 select(date\_renewal\_Year\_Month,churn,id) %>%   
 summarise(n=n()) %>%   
 summarise(n=n()) %>%   
 spread("churn", "n")

d4= d4[-32, ] ## to remove date which has NA  
 d4[is.na(d4)]=0 ## to replace NA's with Zero's  
 d4 = as.data.frame(d4)

names(d4)

## [1] "date\_renewal\_Year\_Month" "0"   
## [3] "1"

colnames(d4) = c("date\_renewal\_Year\_Month","retention","churn")

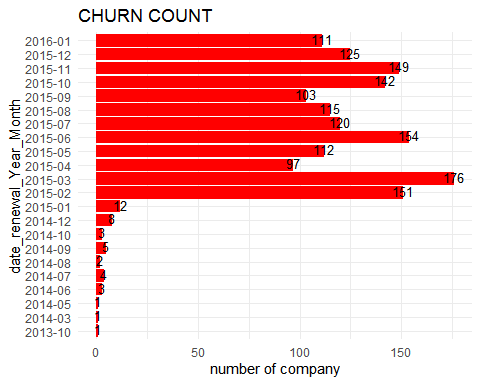
head(d4 %>%   
 mutate(percentage\_churn=`churn`/apply(d4[ ,-1],1,sum)\*100,   
 percentage\_retention=100-percentage\_churn,  
 Total\_no\_company=apply(d4[,-1],1,sum)) %>%   
 select(date\_renewal\_Year\_Month,retention,percentage\_retention,  
 churn,percentage\_churn,Total\_no\_company))

## date\_renewal\_Year\_Month retention percentage\_retention churn percentage\_churn  
## 1 2013-06 1 100.0 0 0.0  
## 2 2013-07 4 100.0 0 0.0  
## 3 2013-08 8 100.0 0 0.0  
## 4 2013-09 4 100.0 0 0.0  
## 5 2013-10 7 87.5 1 12.5  
## 6 2013-11 2 100.0 0 0.0  
## Total\_no\_company  
## 1 1  
## 2 4  
## 3 8  
## 4 4  
## 5 8  
## 6 2

## 

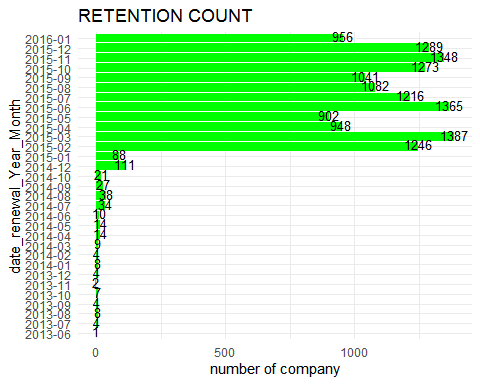
## Visualization for Churn

d4 %>% filter(churn>=1) %>%   
 ggplot(aes(x=date\_renewal\_Year\_Month, y=`churn`)) +  
 geom\_bar(stat="identity", fill="red")+  
 labs(title="CHURN COUNT",x="date\_renewal\_Year\_Month", y= "number of company")+  
 geom\_text(aes(label=churn), vjust=0.3, size=3.5)+  
 theme\_minimal()+coord\_flip()



## Visualization for Retention

d4 %>%   
 ggplot(aes(x=date\_renewal\_Year\_Month, y=retention)) +  
 geom\_bar(stat="identity", fill="green")+  
 labs(title="RETENTION COUNT",x="date\_renewal\_Year\_Month", y= "number of company")+  
 geom\_text(aes(label=retention), vjust=0.3, size=3.5)+  
 theme\_minimal() +  
 coord\_flip()



## Forecast

forecast = train %>%   
 select(id, forecast\_base\_bill\_ele, forecast\_base\_bill\_year, forecast\_bill\_12m, forecast\_cons , forecast\_cons\_12m, forecast\_cons\_year, forecast\_discount\_energy, forecast\_meter\_rent\_12m, forecast\_price\_energy\_p1, forecast\_price\_energy\_p2, forecast\_price\_pow\_p1, churn)  
   
names(forecast)

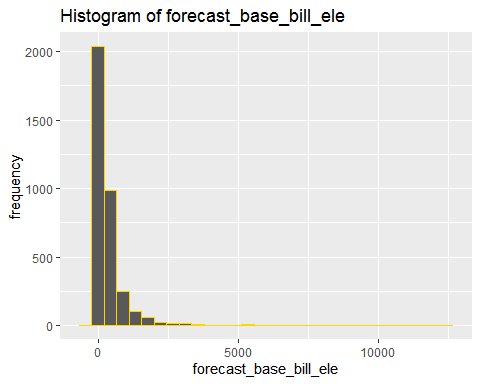
## [1] "id" "forecast\_base\_bill\_ele"   
## [3] "forecast\_base\_bill\_year" "forecast\_bill\_12m"   
## [5] "forecast\_cons" "forecast\_cons\_12m"   
## [7] "forecast\_cons\_year" "forecast\_discount\_energy"  
## [9] "forecast\_meter\_rent\_12m" "forecast\_price\_energy\_p1"  
## [11] "forecast\_price\_energy\_p2" "forecast\_price\_pow\_p1"   
## [13] "churn"

colSums(is.na(forecast))

## id forecast\_base\_bill\_ele forecast\_base\_bill\_year   
## 0 12588 12588   
## forecast\_bill\_12m forecast\_cons forecast\_cons\_12m   
## 12588 12588 0   
## forecast\_cons\_year forecast\_discount\_energy forecast\_meter\_rent\_12m   
## 0 126 0   
## forecast\_price\_energy\_p1 forecast\_price\_energy\_p2 forecast\_price\_pow\_p1   
## 126 126 126   
## churn   
## 0

## Total (retention + churn)

qplot(forecast$forecast\_base\_bill\_ele, geom = "histogram",  
 color = I("GOLD"),  
 xlab = "forecast\_base\_bill\_ele",  
 ylab = "frequency",  
 main = "Histogram of forecast\_base\_bill\_ele")

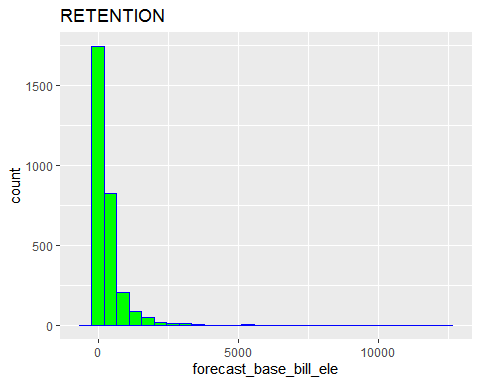


## Histogram for forecast\_base\_bill\_ele (RETENTION)

forecast %>% filter(churn==0) %>%   
 ggplot(aes(forecast\_base\_bill\_ele)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 11524 rows containing non-finite values (stat\_bin).

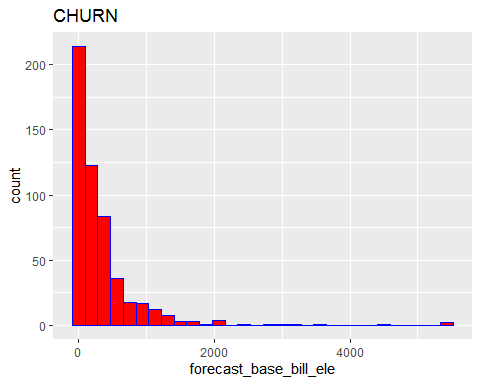


## Histogram for forecast\_base\_bill\_ele (CHURN)

forecast%>% filter(churn==1) %>%   
 ggplot(aes(forecast\_base\_bill\_ele)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1064 rows containing non-finite values (stat\_bin).

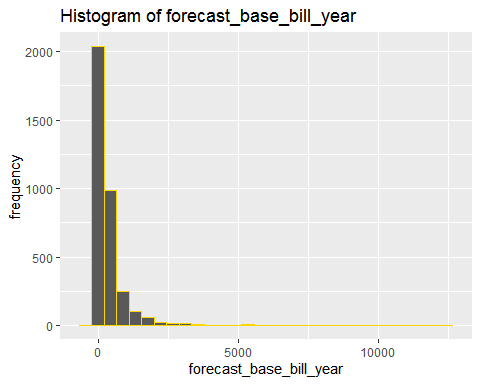


## Total (retention + churn)

qplot(forecast$forecast\_base\_bill\_year, geom = "histogram",  
 color = I("GOLD"),  
 xlab = "forecast\_base\_bill\_year",  
 ylab = "frequency",  
 main = "Histogram of forecast\_base\_bill\_year")

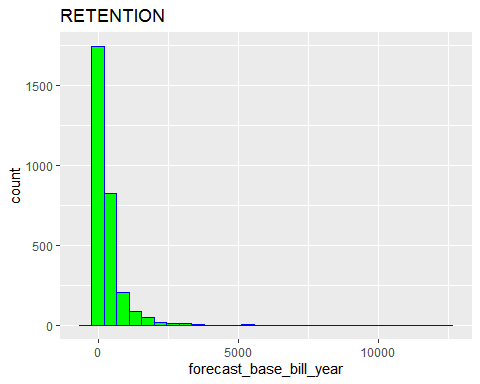
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 12588 rows containing non-finite values (stat\_bin).



## Histogram for forecast\_base\_bill\_year (RETENTION)

forecast %>%   
 filter(churn==0) %>%   
 ggplot(aes(forecast\_base\_bill\_year)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")

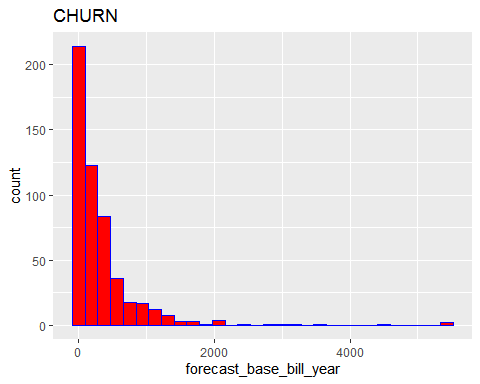


## Histogram for forecast\_base\_bill\_year (CHURN)

forecast %>% filter(churn==1) %>%   
 ggplot(aes(forecast\_base\_bill\_year)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

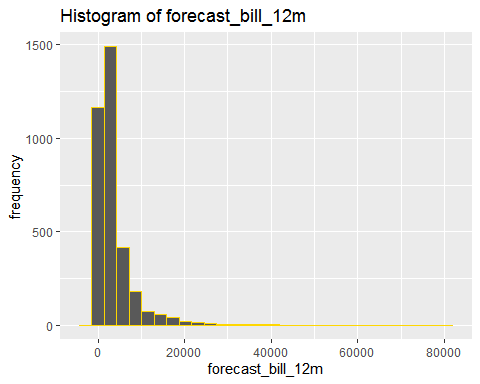
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1064 rows containing non-finite values (stat\_bin).



## Total (retention + churn)

qplot(forecast$forecast\_bill\_12m, geom = "histogram",  
 color = I("GOLD"),  
 xlab = "forecast\_bill\_12m",  
 ylab = "frequency",  
 main = "Histogram of forecast\_bill\_12m")

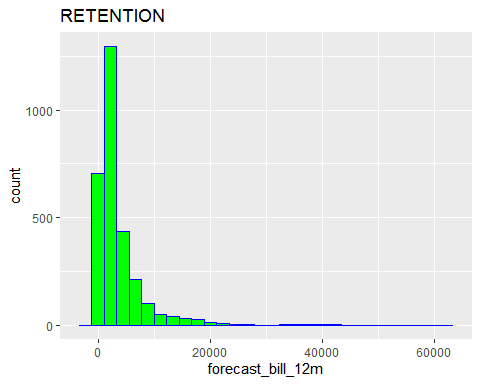


## Histogram for forecast\_bill\_12m (RETENTION)

forecast %>% filter(churn==0) %>%   
 ggplot(aes(forecast\_bill\_12m)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")

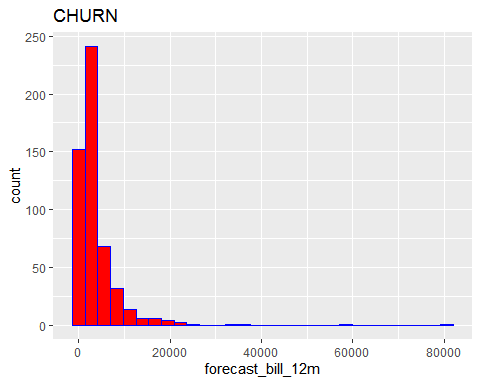
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 11524 rows containing non-finite values (stat\_bin).



## Histogram for forecast\_bill\_12m (CHURN)

forecast %>% filter(churn==1) %>%   
 ggplot(aes(forecast\_bill\_12m)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

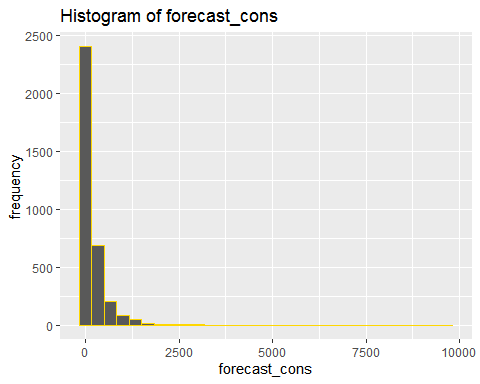


## Total (retention + churn)

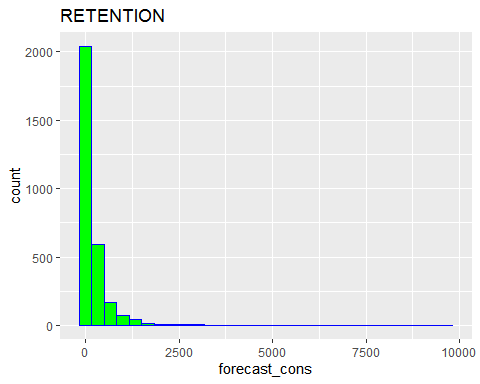
qplot(forecast$forecast\_cons, geom = "histogram",  
 color = I("GOLD"),  
 xlab = "forecast\_cons",  
 ylab = "frequency",  
 main = "Histogram of forecast\_cons")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 12588 rows containing non-finite values (stat\_bin).

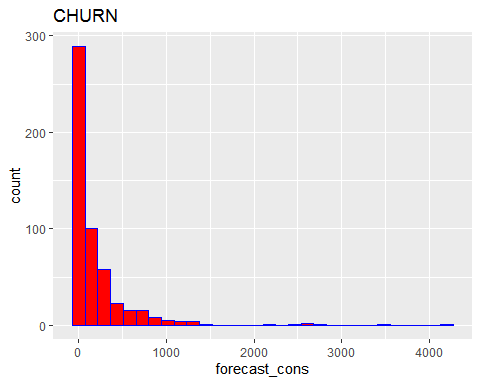


## Histogram for forecast\_cons (RETENTION)

forecast %>% filter(churn==0) %>%   
 ggplot(aes(forecast\_cons)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION") 

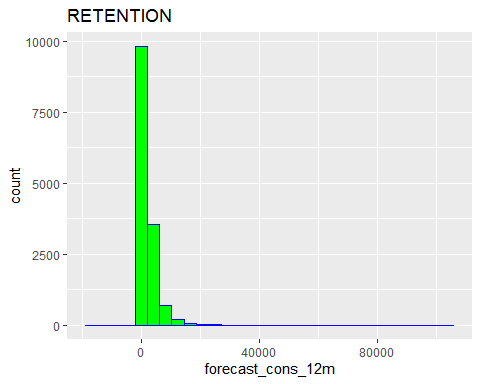
## Histogram for forecast\_cons (CHURN)

forecast %>% filter(churn==1) %>%   
 ggplot(aes(forecast\_cons)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")



## Histogram for forecast\_cons\_12m (RETENTION)

forecast %>%   
 filter(churn==0) %>%   
 ggplot(aes(forecast\_cons\_12m)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")

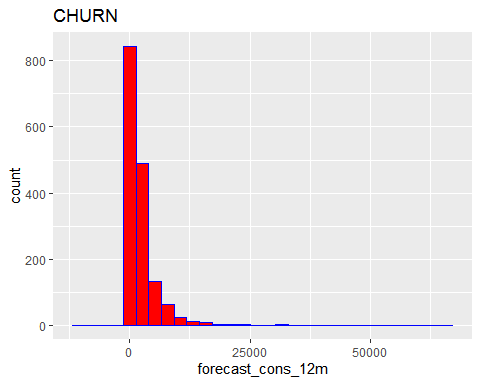


## 

## Histogram for forecast\_cons\_12m (CHURN)

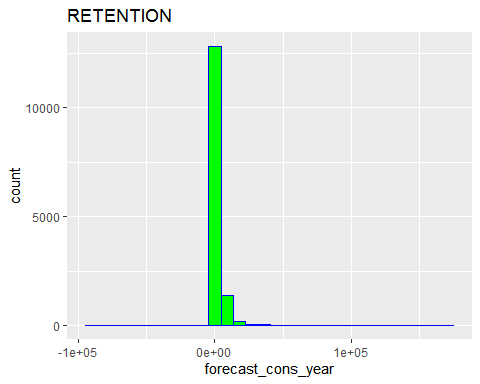
forecast %>% filter(churn==1) %>%   
 ggplot(aes(forecast\_cons\_12m)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



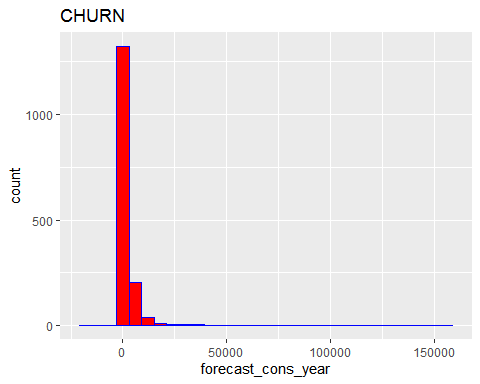
## Histogram for forecast\_cons\_year (RETENTION)

forecast %>%   
 filter(churn==0) %>%   
 ggplot(aes(forecast\_cons\_year)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")



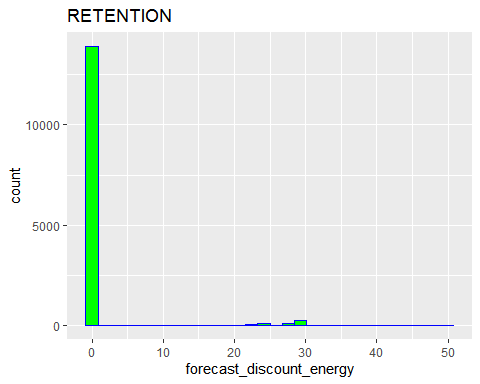
## Histogram for forecast\_cons\_year (CHURN)

forecast %>%   
 filter(churn==1) %>%   
 ggplot(aes(forecast\_cons\_year)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")



## Histogram for forecast\_discount\_energy (RETENTION)

forecast %>%   
 filter(churn==0) %>%   
 ggplot(aes(forecast\_discount\_energy)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")

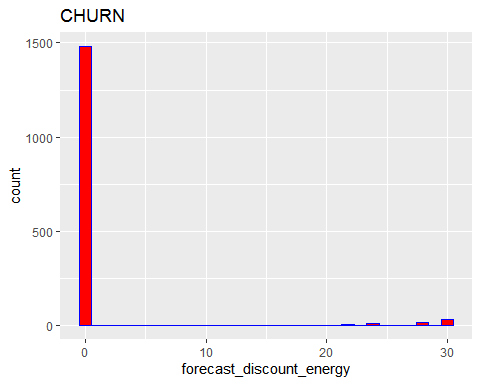


## Histogram for forecast\_discount\_energy (CHURN)

forecast %>%   
 filter(churn==1) %>%   
 ggplot(aes(forecast\_discount\_energy)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

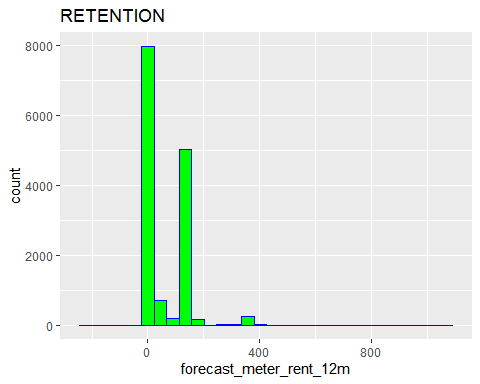
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 46 rows containing non-finite values (stat\_bin).



## Histogram for forecast\_meter\_rent\_12m (RETENTION)

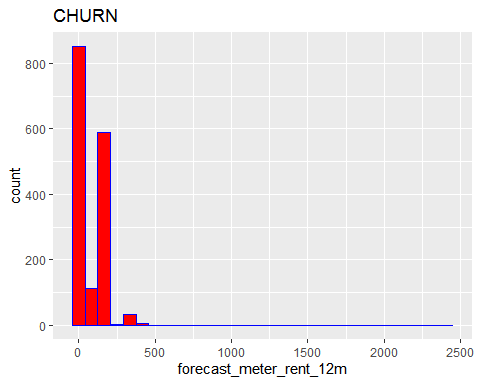
forecast %>%   
 filter(churn==0) %>%   
 ggplot(aes(forecast\_meter\_rent\_12m)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")



## Histogram for forecast\_meter\_rent\_12m (CHURN)

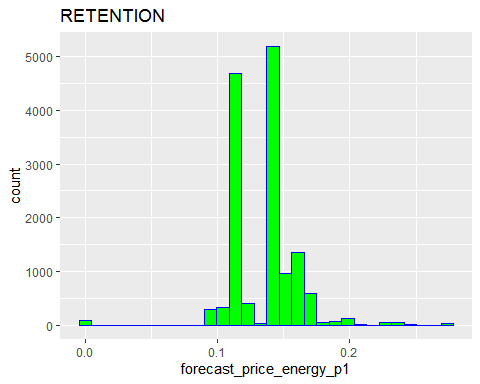
forecast %>% filter(churn==1) %>%   
 ggplot(aes(forecast\_meter\_rent\_12m)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Histogram for forecast\_price\_energy\_p1 (RETENTION)

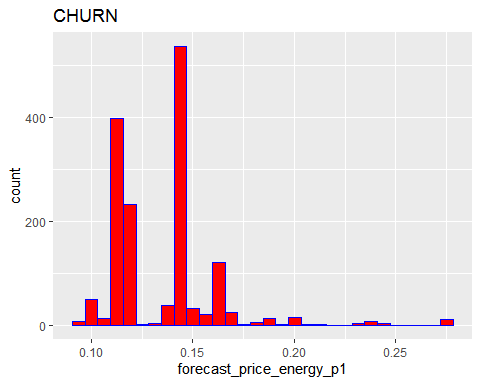
forecast %>%   
 filter(churn==0) %>%   
 ggplot(aes(forecast\_price\_energy\_p1)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")



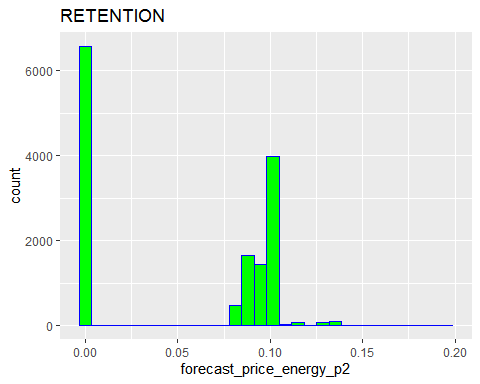
## 

## Histogram for forecast\_price\_energy\_p1 (CHURN)

forecast %>% filter(churn==1) %>%   
 ggplot(aes(forecast\_price\_energy\_p1)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")

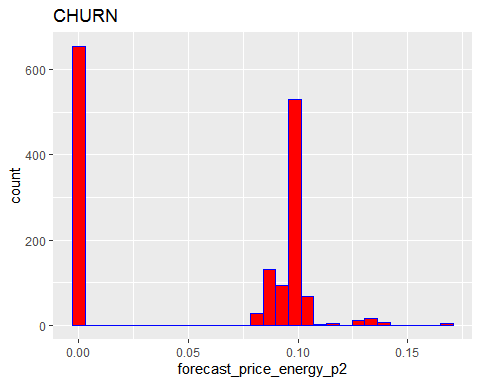


## Histogram for forecast\_price\_energy\_p2 (RETENTION)

forecast %>%   
 filter(churn==0) %>%   
 ggplot(aes(forecast\_price\_energy\_p2)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION") 

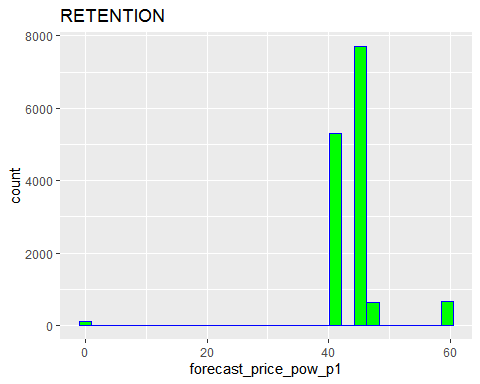
## Histogram for forecast\_price\_energy\_p2 (CHURN)

forecast %>% filter(churn==1) %>%   
 ggplot(aes(forecast\_price\_energy\_p2)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")



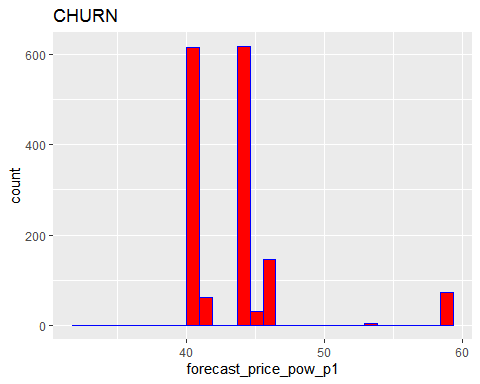
## Histogram for forecast\_price\_pow\_p1 (RETENTION)

forecast %>%   
 filter(churn==0) %>%   
 ggplot(aes(forecast\_price\_pow\_p1)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")



## Histogram for forecast\_price\_pow\_p1 (CHURN)

forecast %>% filter(churn==1) %>%   
 ggplot(aes(forecast\_price\_pow\_p1)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")



## CONTRACT\_TYPE

contract\_type = train %>%   
 select(id, has\_gas, churn)  
 class(contract\_type)

## [1] "data.frame"

## 

## BAR PLOT CONTRACT\_TYPE CHURN

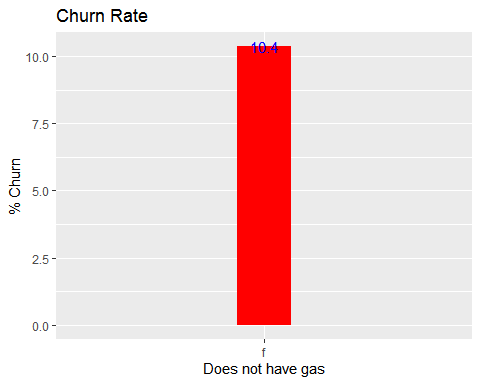
contract\_plot = contract\_type %>%   
 group\_by(churn,has\_gas) %>%  
 summarise(n=n()) %>%   
 spread("churn","n")

colnames(contract\_plot) = c("has\_gas","retention","churn")

contract\_plot = contract\_plot %>%   
 mutate(churn\_rate = churn/rowSums(contract\_plot[ ,-1])\*100,  
 retention\_rate = 100-churn\_rate, total\_no = rowSums(contract\_plot[ ,-1])) %>%   
 select(has\_gas,retention,retention\_rate,churn,churn\_rate, total\_no)

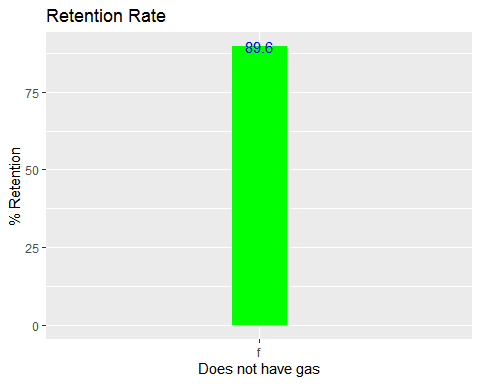
## Visualization for has\_gas= "f" which Churned

contract\_plot %>% filter(has\_gas=="f") %>%   
 ggplot(aes(has\_gas,churn\_rate))+  
 geom\_col(position="dodge", fill="red")+  
 labs(title= "Churn Rate",  
 x="Does not have gas",   
 y= "% Churn")+  
 geom\_text(aes(label = round(churn\_rate,1)),  
 position = position\_dodge(7),  
 color="blue",vjust = 0.5,hjust = 0.5)

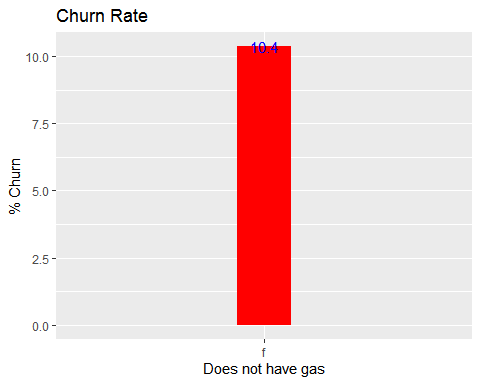
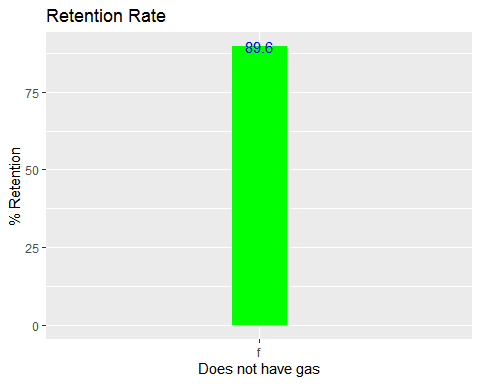


## Visualization for has\_gas= "f" which Retained

contract\_plot %>%   
 filter(has\_gas=="f") %>%   
 ggplot(aes(has\_gas,retention\_rate))+  
 geom\_col(position="dodge", fill = "green")+  
 labs(title= "Retention Rate",  
 x="Does not have gas",   
 y= "% Retention")+  
 geom\_text(aes(label = round(retention\_rate,1)),  
 position = position\_dodge(7),   
 color="blue",vjust = 0.5,hjust = 0.5)



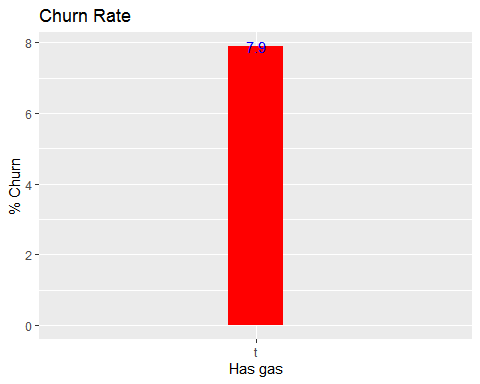
COMPARISION BETWEEN RETENTION AND CHURN RATES WHICH DOES NOT “HAVE\_GAS”



## 

## Visualization for has\_gas= "t" which Churned

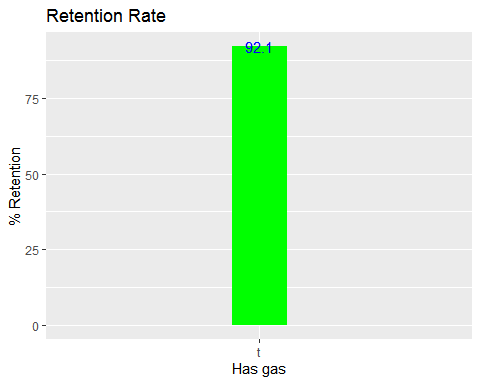
contract\_plot %>%   
 filter(has\_gas=="t") %>%   
 ggplot(aes(has\_gas,churn\_rate))+  
 geom\_col(position="dodge", fill="red")+  
 labs(title= "Churn Rate",  
 x="Has gas",   
 y= "% Churn")+  
 geom\_text(aes(label = round(churn\_rate,1)),  
 position = position\_dodge(7),  
 color="blue",vjust = 0.5,hjust = 0.5)



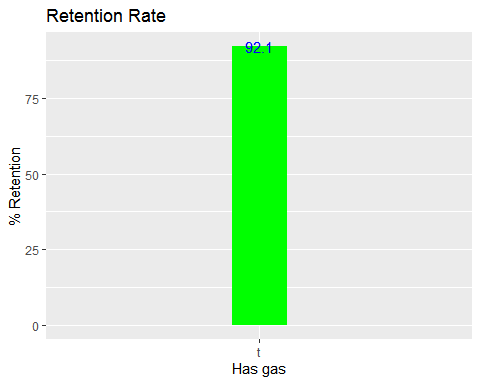
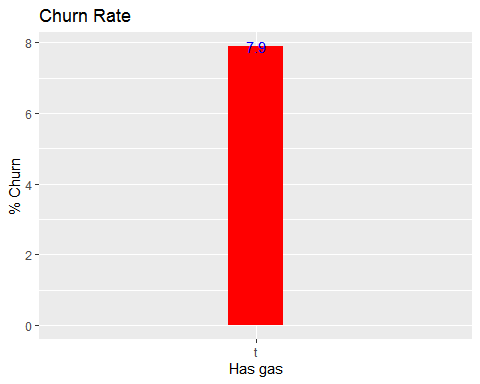
## 

## Visualization for has\_gas = "t" which Retained

contract\_plot %>% filter(has\_gas=="t") %>%   
 ggplot(aes(has\_gas,retention\_rate))+  
 geom\_col(position="dodge", fill="green")+  
 labs(title= "Retention Rate",  
 x="Has gas",   
 y= "% Retention")+  
 geom\_text(aes(label = round(retention\_rate,1)),  
 position = position\_dodge(7),  
 color="blue",vjust = 0.5,hjust = 0.5)



COMPARISION BETWEEN RETENTION AND CHURN RATES WHICH“HAS\_GAS”

## 

## MARGINS

margins = train %>%   
 select(id, margin\_gross\_pow\_ele,margin\_net\_pow\_ele,net\_margin)  
 colSums(is.na(margins))

## id margin\_gross\_pow\_ele margin\_net\_pow\_ele   
## 0 13 13   
## net\_margin   
## 15

## Box Plots

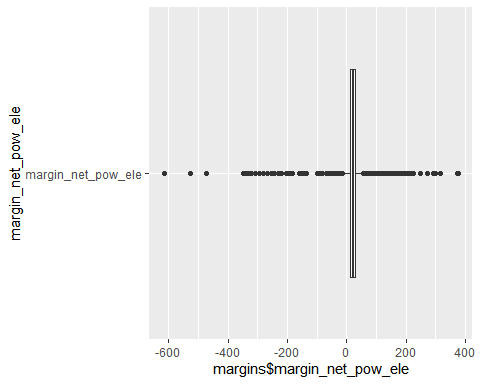
qplot("margin\_gross\_pow\_ele", margins$margin\_gross\_pow\_ele, geom = "boxplot") + coord\_flip()

## Warning: Removed 13 rows containing non-finite values (stat\_boxplot).



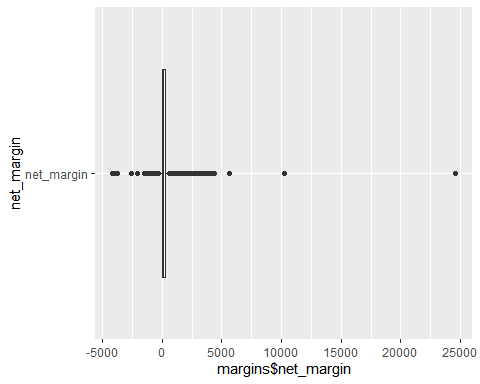
qplot("margin\_net\_pow\_ele", margins$margin\_net\_pow\_ele, geom = "boxplot") + coord\_flip()

## Warning: Removed 13 rows containing non-finite values (stat\_boxplot).



qplot("net\_margin",margins$net\_margin, geom = "boxplot") + coord\_flip()

## Warning: Removed 15 rows containing non-finite values (stat\_boxplot).



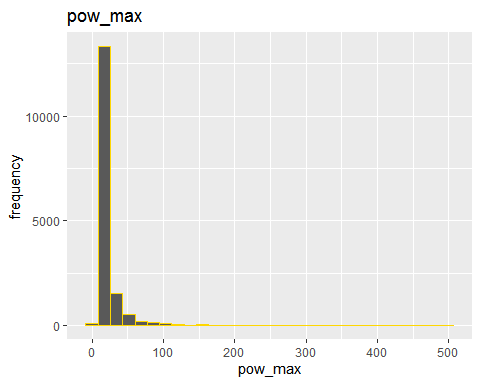
## SUBSCRIBED POWER

power = train %>%   
 select(id, pow\_max, churn)  
   
colSums(is.na(power))

## id pow\_max churn   
## 0 3 0

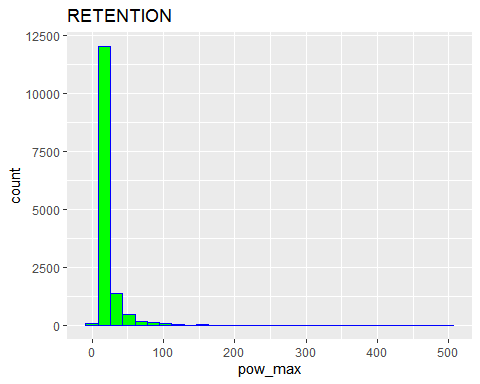
## Total (retention + churn)

qplot(power$pow\_max, geom = "histogram",  
 color = I("GOLD"),  
 xlab = "pow\_max",  
 ylab = "frequency",  
 main = "pow\_max")



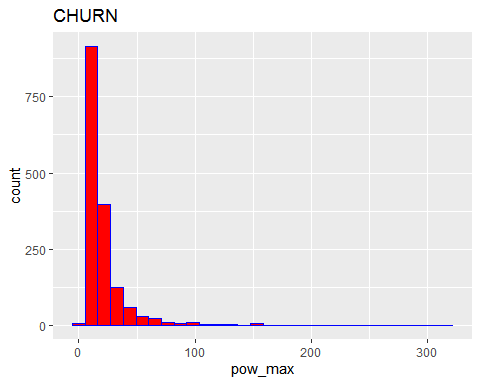
## Histogram for pow\_max (RETENTION)

power %>% filter(churn==0) %>%   
 ggplot(aes(pow\_max)) +   
 geom\_histogram(fill="green",color = I("blue")) +   
 ggtitle("RETENTION")



## Histogram for forecast\_base\_bill\_ele (CHURN)

power %>% filter(churn==1) %>%   
 ggplot(aes(pow\_max)) +   
 geom\_histogram(fill="red",color = I("blue")) +   
 ggtitle("CHURN")



## OTHERS

others = train %>%   
 select(id, nb\_prod\_act, num\_years\_antig, origin\_up, churn)  
 glimpse(others)

## Rows: 16,096  
## Columns: 5  
## $ id <chr> "0002203ffbb812588b632b9e628cc38d", "0004351ebdd665...  
## $ nb\_prod\_act <int> 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 1, ...  
## $ num\_years\_antig <int> 6, 6, 3, 6, 6, 4, 3, 3, 12, 3, 6, 5, 7, 4, 7, 4, 3,...  
## $ origin\_up <chr> "kamkkxfxxuwbdslkwifmmcsiusiuosws", "kamkkxfxxuwbds...  
## $ churn <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, ...

## nb\_prod\_act

others\_1 = others %>%   
 group\_by(nb\_prod\_act,churn,id) %>%  
 select(nb\_prod\_act,churn,id) %>%   
 summarise(n=n()) %>%   
 summarise(n=n()) %>%   
 spread("churn", "n")

## `summarise()` regrouping output by 'nb\_prod\_act', 'churn' (override with `.groups` argument)

## `summarise()` regrouping output by 'nb\_prod\_act' (override with `.groups` argument)

others\_1[is.na(others\_1)]=0 ## to replace NA's with Zero's  
   
 class(others\_1)

## [1] "grouped\_df" "tbl\_df" "tbl" "data.frame"

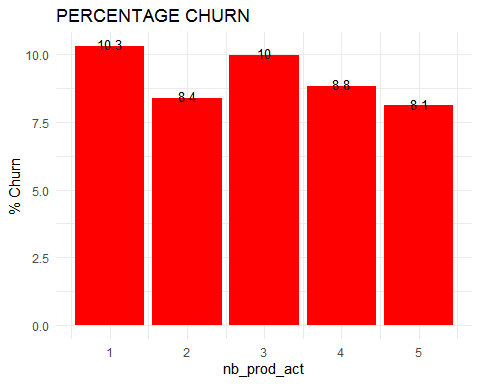
others\_1 = as.data.frame(others\_1)  
 colnames(others\_1) = c("nb\_prod\_act","retention","churn")

others\_1 = others\_1 %>% mutate(percentage\_churn = churn/rowSums(others\_1[ ,-1])\*100,   
 percentage\_retention=100-percentage\_churn,  
 Total\_no\_company = rowSums(others\_1[,-1])) %>%   
 select(nb\_prod\_act,retention,percentage\_retention,  
 churn,percentage\_churn,Total\_no\_company)

## 

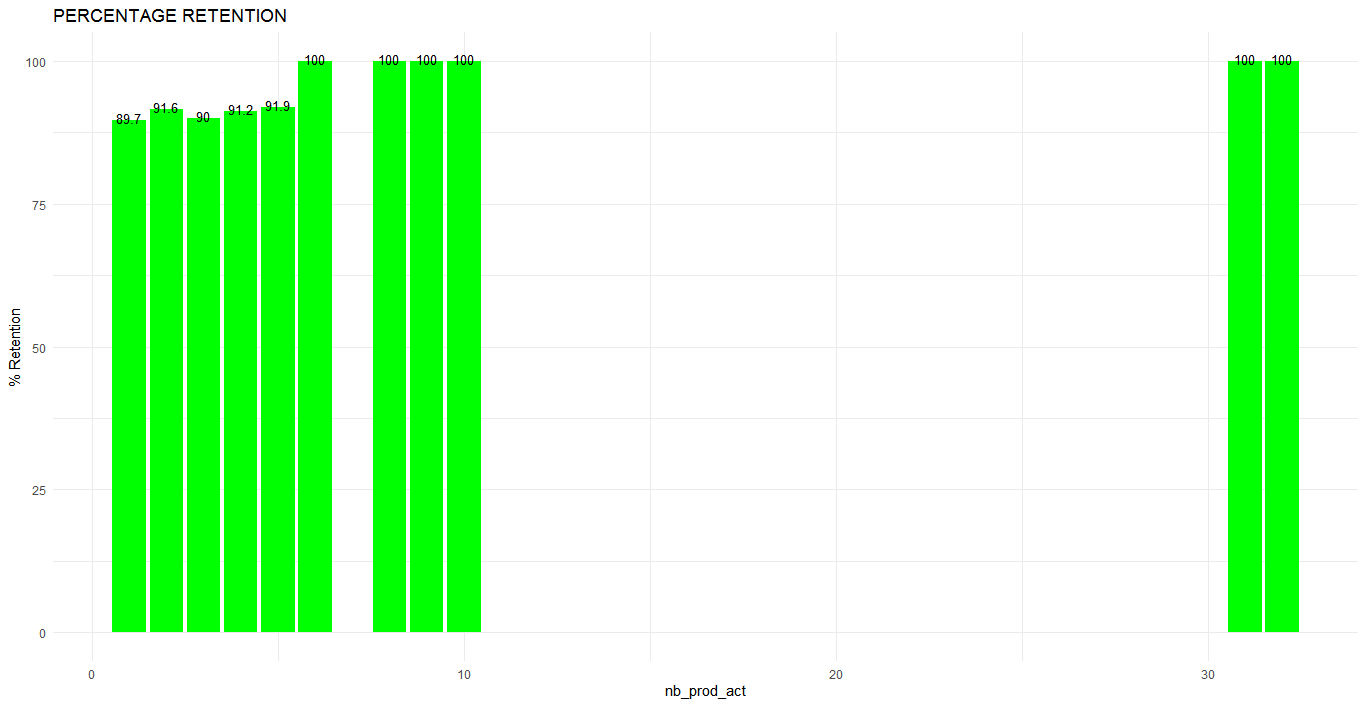
## Visualization for Churn

others\_1 %>%   
 filter(percentage\_churn>=1) %>%   
 ggplot(aes(x= nb\_prod\_act, y= percentage\_churn)) +  
 geom\_bar(stat="identity", fill="red")+  
 labs(title="PERCENTAGE CHURN",x="nb\_prod\_act", y= "% Churn")+  
 geom\_text(aes(label= round(percentage\_churn,1)), ## You can add"position = position\_dodge(1)"  
 vjust=0.3, size=3.5)+ ## to adjust size of bars  
 theme\_minimal()



## Visualization for Retention

others\_1 %>%   
 ggplot(aes(x=nb\_prod\_act, y= percentage\_retention)) +  
 geom\_bar(stat="identity", fill="green")+  
 labs(title="PERCENTAGE RETENTION",x="nb\_prod\_act", y= "% Retention")+  
 geom\_text(aes(label= round(percentage\_retention,1)), vjust=0.3, size=3.5)+  
 theme\_minimal()



## num\_years\_antig

others\_2 = others %>%   
 group\_by(num\_years\_antig,churn,id) %>%  
 select(num\_years\_antig,churn,id) %>%   
 summarise(n=n()) %>%   
 summarise(n=n()) %>%   
 spread("churn", "n")

## `summarise()` regrouping output by 'num\_years\_antig', 'churn' (override with `.groups` argument)

## `summarise()` regrouping output by 'num\_years\_antig' (override with `.groups` argument)

others\_2[is.na(others\_2)]=0 ## to replace NA's with Zero's  
 class(others\_2)

## [1] "grouped\_df" "tbl\_df" "tbl" "data.frame"

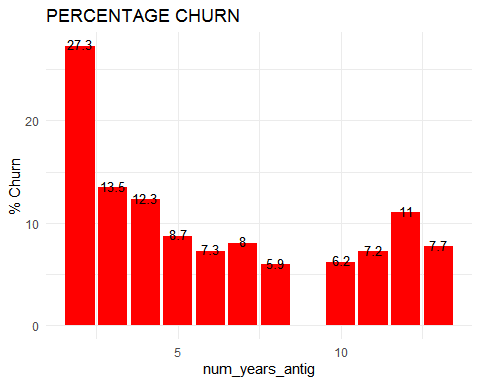
others\_2 = as.data.frame(others\_2)  
   
 colnames(others\_2) = c("num\_years\_antig","retention","churn")

others\_2 = others\_2 %>% mutate(percentage\_churn = churn/rowSums(others\_2[ ,-1])\*100,   
 percentage\_retention=100-percentage\_churn,  
 Total\_no\_company = rowSums(others\_2[,-1])) %>%   
 select(num\_years\_antig,retention,percentage\_retention,  
 churn,percentage\_churn,Total\_no\_company)

## 

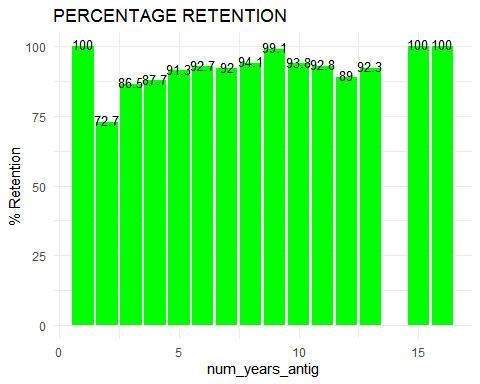
## Visualization for Churn

others\_2 %>% filter(percentage\_churn>=1) %>%   
 ggplot(aes(x= num\_years\_antig, y= percentage\_churn)) +  
 geom\_bar(stat="identity", fill="red")+  
 labs(title="PERCENTAGE CHURN",x="num\_years\_antig", y= "% Churn")+  
 geom\_text(aes(label= round(percentage\_churn,1)), ## You can add"position = position\_dodge(1)"  
 vjust=0.3, size=3.5)+ ## to adjust size of bars  
 theme\_minimal()



## Visualization for Retention

others\_2 %>%   
 ggplot(aes(x=num\_years\_antig, y= percentage\_retention)) +  
 geom\_bar(stat="identity", fill="green")+  
 labs(title="PERCENTAGE RETENTION",x="num\_years\_antig", y= "% Retention")+  
 geom\_text(aes(label= round(percentage\_retention,1)), vjust=0.3, size=3.5)+  
 theme\_minimal()



## origin\_up

others\_3 = others %>%   
 group\_by(origin\_up,churn,id) %>%  
 select(origin\_up,churn,id) %>%   
 summarise(n=n()) %>%   
 summarise(n=n()) %>%   
 spread("churn", "n")

others\_3 = others\_3[-1, ] ## to remove rows  
 others\_3[is.na(others\_3)]=0 ## to replace NA's with Zero's  
 class(others\_3)

## [1] "grouped\_df" "tbl\_df" "tbl" "data.frame"

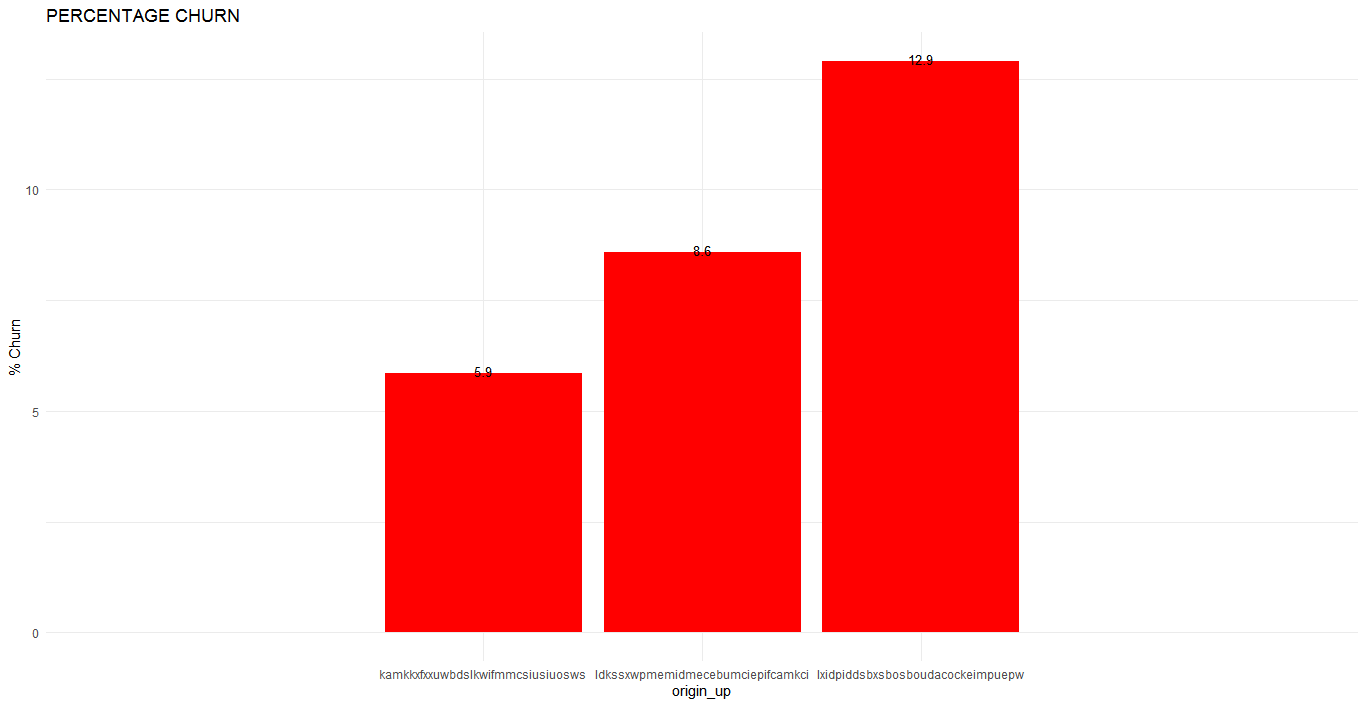
others\_3 = as.data.frame(others\_3)  
   
 colnames(others\_3) = c("origin\_up","retention","churn")

others\_3 = others\_3 %>% mutate(percentage\_churn = churn/rowSums(others\_3[ ,-1])\*100,

percentage\_retention=100-percentage\_churn,  
 total\_no\_company = rowSums(others\_3[,-1])) %>%   
 select(origin\_up,retention,percentage\_retention,  
 churn,percentage\_churn,total\_no\_company)

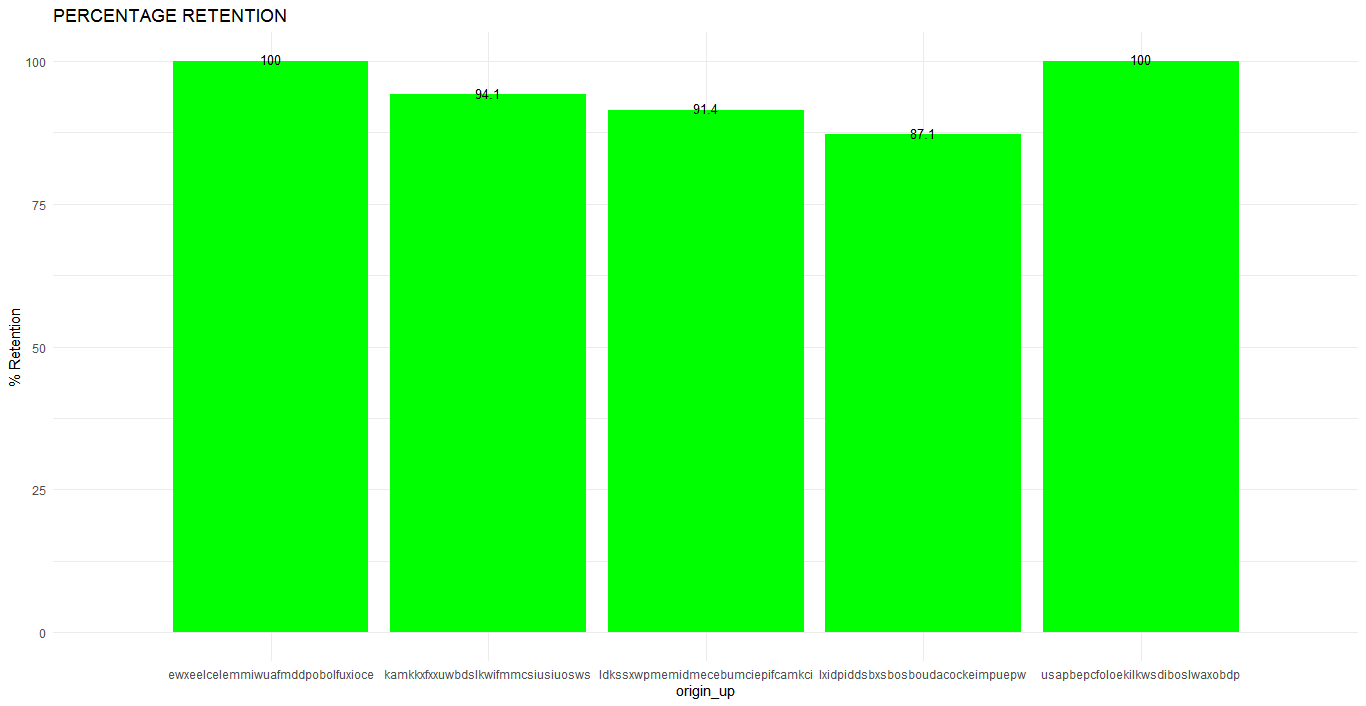
## Bar Plot Visualization for Churn

others\_3 %>%   
 filter(percentage\_churn>=1) %>%   
 ggplot(aes(x= origin\_up, y= percentage\_churn)) +  
 geom\_bar(stat="identity", fill="red")+  
 labs(title="PERCENTAGE CHURN",x="origin\_up", y= "% Churn")+  
 geom\_text(aes(label= round(percentage\_churn,1)), ## You can add"position = position\_dodge(1)"  
 vjust=0.3, size=3.5, position=position\_dodge(4))+ ## to adjust size of bars  
 theme\_minimal()



## Barplot Visualization for Retention

others\_3 %>%   
 ggplot(aes(x=origin\_up, y= percentage\_retention)) +  
 geom\_bar(stat="identity", fill="green")+  
 labs(title="PERCENTAGE RETENTION",x="origin\_up", y= "% Retention")+  
 geom\_text(aes(label= round(percentage\_retention,1)),   
 vjust=0.3, size=3.5, position=position\_dodge(2))+  
 theme\_minimal()



## DATA CLEANING

## Missing Values in train data set

train\_2 = train

## Changing dates into date format

train\_2$date\_activ = as.Date(train\_2$date\_activ)  
 train\_2$date\_end = as.Date(train\_2$date\_end)  
 train\_2$date\_modif\_prod = as.Date(train\_2$date\_modif\_prod)  
 train\_2$date\_renewal = as.Date(train\_2$date\_renewal)

missing\_data = apply(train\_2, 2, function(col)sum(is.na(col))/length(col)\*100)   
 class(missing\_data)

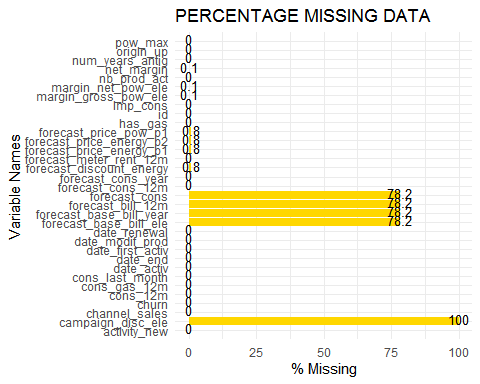
## [1] "numeric"

missing\_data = as.data.frame(missing\_data)

## Plot of missing Data for training data

missing\_data\_2 %>%   
 ggplot(aes(x=Names, y= missing\_data)) +  
 geom\_bar(stat="identity", fill="GOLD")+  
 labs(title="PERCENTAGE MISSING DATA",x="Variable Names", y= "% Missing")+  
 geom\_text(aes(label= round(missing\_data,1)),   
 vjust=0.3, size=3.5, position=position\_dodge(2))+  
 theme\_minimal() + coord\_flip()

## Warning: position\_dodge requires non-overlapping x intervals



## Removal of variables with more than 60% missing values

names(train\_2)

## [1] "id" "activity\_new"   
## [3] "campaign\_disc\_ele" "channel\_sales"   
## [5] "cons\_12m" "cons\_gas\_12m"   
## [7] "cons\_last\_month" "date\_activ"   
## [9] "date\_end" "date\_first\_activ"   
## [11] "date\_modif\_prod" "date\_renewal"   
## [13] "forecast\_base\_bill\_ele" "forecast\_base\_bill\_year"   
## [15] "forecast\_bill\_12m" "forecast\_cons"   
## [17] "forecast\_cons\_12m" "forecast\_cons\_year"   
## [19] "forecast\_discount\_energy" "forecast\_meter\_rent\_12m"   
## [21] "forecast\_price\_energy\_p1" "forecast\_price\_energy\_p2"  
## [23] "forecast\_price\_pow\_p1" "has\_gas"   
## [25] "imp\_cons" "margin\_gross\_pow\_ele"   
## [27] "margin\_net\_pow\_ele" "nb\_prod\_act"   
## [29] "net\_margin" "num\_years\_antig"   
## [31] "origin\_up" "pow\_max"   
## [33] "churn"

train\_2 = train\_2 %>% select(-3,-10,-13:-16)

## 

## Checking for Duplicates

train\_2[duplicated(train\_2)] ## for extracting duplicates

## data frame with 0 columns and 16096 rows

## 

## MISSING DATES

train\_3 = train\_2  
   
 caseDay = ymd("2016-07-30")  
 caseDay\_1 <- ymd("2013-05-01")  
 caseDay\_2 = ymd("2015-07-24")

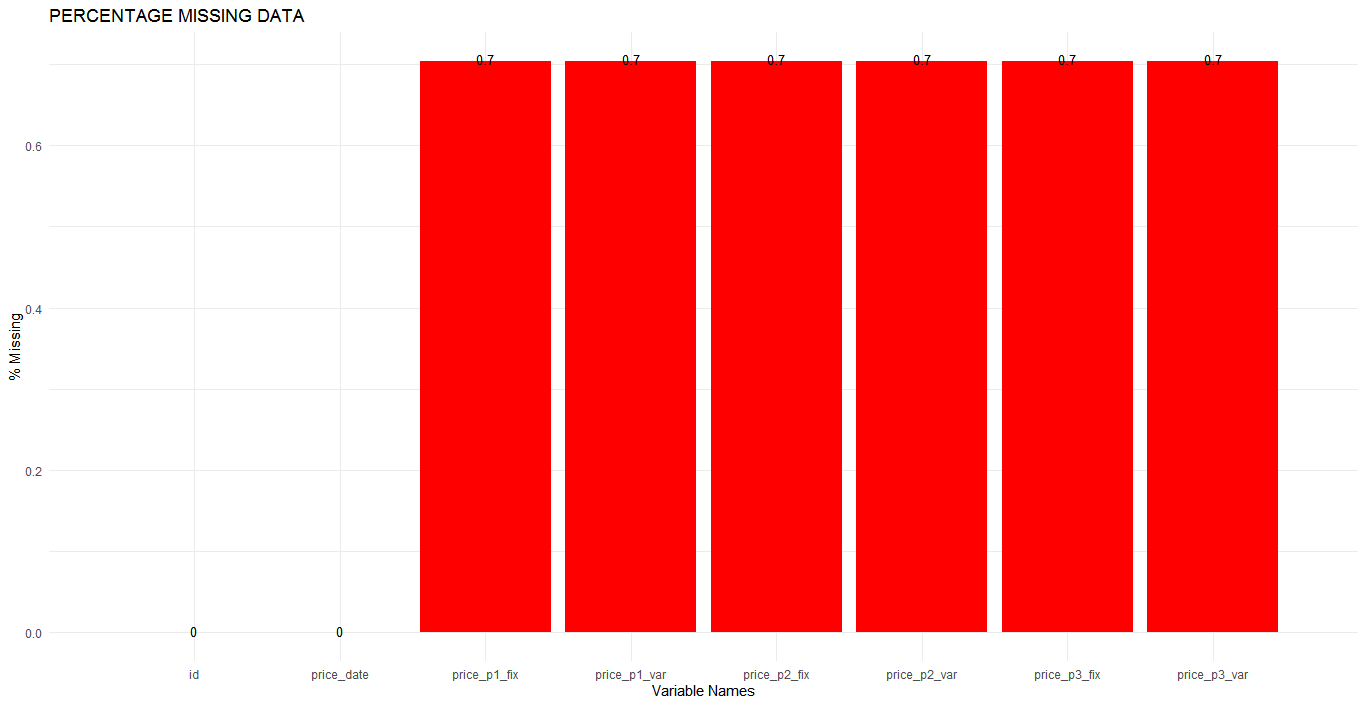
train\_3 = train\_3 %>%  
 mutate(date\_end\_complete = case\_when(is.na(date\_end) ~ caseDay,TRUE ~ date\_end),  
 date\_modif\_prod\_complete = case\_when(is.na(date\_modif\_prod) ~ caseDay\_1,TRUE ~ date\_modif\_prod),  
 date\_renewal\_complete =case\_when(is.na(date\_renewal) ~ caseDay\_2,TRUE ~ date\_renewal))

## Missing Data for Pricing\_data

percent\_missing\_pricing\_data = apply(pricing\_data,2, function(col) sum(is.na(col))/length(col)\*100)  
   
 percent\_missing\_pricing\_data = read.csv("percent\_missing\_pricing\_data.csv")  
 colnames(percent\_missing\_pricing\_data) = c("variable\_name", "percentage")

## Visualization for missing data in pricing\_data

percent\_missing\_pricing\_data %>%   
 ggplot(aes(x=variable\_name, y= percentage)) +  
 geom\_bar(stat="identity", fill="red")+  
 labs(title="PERCENTAGE MISSING DATA",x="Variable Names", y= "% Missing")+  
 geom\_text(aes(label= round(percentage,1)),   
 vjust=0.3, size=3.5, position=position\_dodge(2))+  
 theme\_minimal()



## Since very little data is missing in the pricing\_data,we simply replace each missing values with their RESPECTIVE median (i.e Column-wise)

p\_1 = pricing\_data  
 colSums(is.na(p\_1))

## id price\_date price\_p1\_var price\_p2\_var price\_p3\_var price\_p1\_fix   
## 0 0 1359 1359 1359 1359   
## price\_p2\_fix price\_p3\_fix   
## 1359 1359

p\_1$price\_date = as\_date(p\_1$price\_date) ## Using lubridate package which makes it easy to manipulate dates

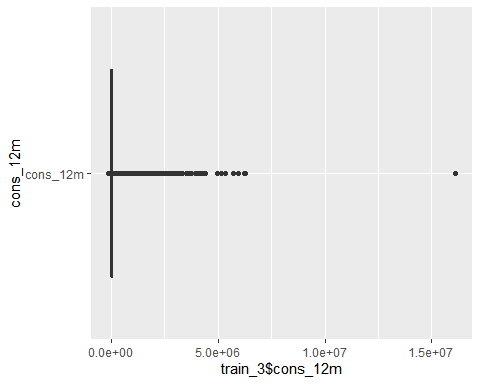
## Assigning the median to respective variables

var1Case = median(na.omit(p\_1)$price\_p1\_var)  
 var2Case = median(na.omit(p\_1)$price\_p2\_var)  
 var3Case = median(na.omit(p\_1)$price\_p3\_var)  
   
 fix1Case = median(na.omit(p\_1)$price\_p1\_fix)  
 fix2Case = median(na.omit(p\_1)$price\_p2\_fix)  
 fix3Case = median(na.omit(p\_1)$price\_p2\_fix)

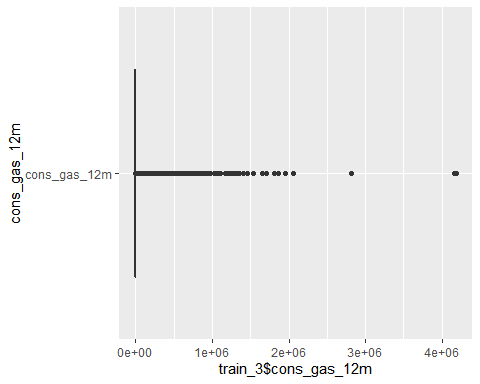
## Replacement of missing values with median takes place

p\_1 = p\_1 %>%  
 mutate(price\_p1\_var\_complete = case\_when(is.na(price\_p1\_var) ~ var1Case,TRUE ~ price\_p1\_var),  
 price\_p2\_var\_complete = case\_when(is.na(price\_p2\_var) ~ var2Case,TRUE ~ price\_p2\_var),  
 price\_p3\_var\_complete = case\_when(is.na(price\_p3\_var) ~ var3Case,TRUE ~ price\_p3\_var),  
 price\_p1\_fix\_complete = case\_when(is.na(price\_p1\_fix) ~ fix1Case,TRUE ~ price\_p1\_fix),  
 price\_p2\_fix\_complete = case\_when(is.na(price\_p2\_fix) ~ fix2Case,TRUE ~ price\_p2\_fix),  
 price\_p3\_fix\_complete = case\_when(is.na(price\_p3\_fix) ~ fix3Case,TRUE ~ price\_p3\_fix))

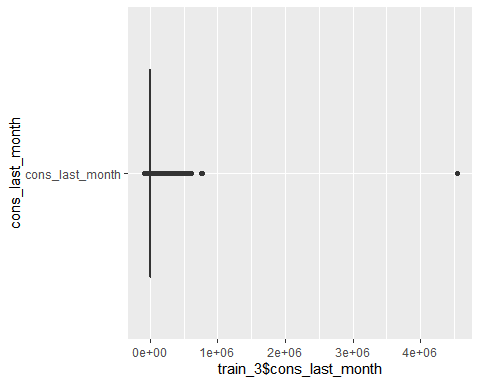
## Box Plots

qplot("cons\_12m", train\_3$cons\_12m, geom = "boxplot") + coord\_flip()

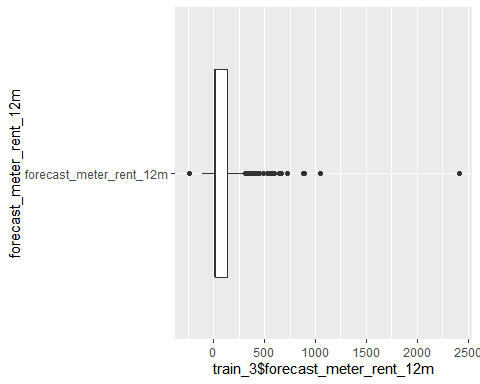
qplot("cons\_gas\_12m", train\_3$cons\_gas\_12m, geom = "boxplot") + coord\_flip()



qplot("cons\_last\_month", train\_3$cons\_last\_month, geom = "boxplot") + coord\_flip()



qplot("forecast\_meter\_rent\_12m", train\_3$forecast\_meter\_rent\_12m,geom = "boxplot") + coord\_flip()



## Removing negative values in the Pricing DataSet

names(p\_1)

## [1] "id" "price\_date" "price\_p1\_var"   
## [4] "price\_p2\_var" "price\_p3\_var" "price\_p1\_fix"   
## [7] "price\_p2\_fix" "price\_p3\_fix" "price\_p1\_var\_complete"  
## [10] "price\_p2\_var\_complete" "price\_p3\_var\_complete" "price\_p1\_fix\_complete"  
## [13] "price\_p2\_fix\_complete" "price\_p3\_fix\_complete"

apply(p\_1 %>% select(9:14),2,mean)

## price\_p1\_var\_complete price\_p2\_var\_complete price\_p3\_var\_complete   
## 0.14102697 0.05463040 0.03049601   
## price\_p1\_fix\_complete price\_p2\_fix\_complete price\_p3\_fix\_complete   
## 43.33217484 10.62287069 6.40998132

apply(p\_1 %>% select(9:14),2,sd)

## price\_p1\_var\_complete price\_p2\_var\_complete price\_p3\_var\_complete   
## 0.02503241 0.04992426 0.03629801   
## price\_p1\_fix\_complete price\_p2\_fix\_complete price\_p3\_fix\_complete   
## 5.41934469 12.84189866 7.77359458

apply(p\_1 %>% select(9:14),2,min)

## price\_p1\_var\_complete price\_p2\_var\_complete price\_p3\_var\_complete   
## 0.0000000 0.0000000 0.0000000   
## price\_p1\_fix\_complete price\_p2\_fix\_complete price\_p3\_fix\_complete   
## -0.1777788 -0.0977520 -0.0651720

apply(p\_1 %>% select(9:14),2,max)

## price\_p1\_var\_complete price\_p2\_var\_complete price\_p3\_var\_complete   
## 0.280700 0.229788 0.114102   
## price\_p1\_fix\_complete price\_p2\_fix\_complete price\_p3\_fix\_complete   
## 59.444710 36.490692 17.458221

apply(p\_1 %>% select(9:14),2,quantile, c(0.25,0.50,0.75))

## price\_p1\_var\_complete price\_p2\_var\_complete price\_p3\_var\_complete  
## 25% 0.125976 0.000000 0.000000  
## 50% 0.146033 0.085483 0.000000  
## 75% 0.151635 0.101673 0.072558  
## price\_p1\_fix\_complete price\_p2\_fix\_complete price\_p3\_fix\_complete  
## 25% 40.72888 0.00000 0.00000  
## 50% 44.26693 0.00000 0.00000  
## 75% 44.44471 24.33958 16.22639

## Investigating how many negative values exist

head( p\_1 %>%   
 select(price\_p1\_fix\_complete) %>%   
 arrange(price\_p1\_fix\_complete),15L)

## price\_p1\_fix\_complete  
## 1 -0.1777788  
## 2 -0.1629156  
## 3 -0.1629156  
## 4 -0.1629156  
## 5 -0.1629156  
## 6 -0.1629156  
## 7 -0.1629156  
## 8 -0.1629156  
## 9 -0.1629120  
## 10 -0.1629120  
## 11 0.0000000  
## 12 0.0000000  
## 13 0.0000000  
## 14 0.0000000  
## 15 0.0000000

which(p\_1$price\_p1\_fix\_complete<0)

## [1] 23139 28351 98576 113468 118468 125820 128762 141012 160828 181812

head(p\_1[c(23139,28351,98576, 113468, 118468, 125820, 128762, 141012, 160828, 181812), ],5L)

## id price\_date price\_p1\_var price\_p2\_var  
## 23139 951d99fe07ca94c2139f43bc37095139 2001-03-15 0.125976 0.103395  
## 28351 f7bdc6fa1067cd26fd80bfb9f3fca28f 2001-03-15 0.131032 0.108896  
## 98576 9b523ad5ba8aa2e524dcda5b3d54dab2 2001-02-15 0.129444 0.106863  
## 113468 cfd098ee6c567eb32374c77d20571bc7 2001-02-15 0.123086 0.100505  
## 118468 51d7d8a0bf6b8bd94f8c1de7942c66ea 2001-07-15 0.128132 0.105996  
## price\_p3\_var price\_p1\_fix price\_p2\_fix price\_p3\_fix  
## 23139 0.071536 -0.1629156 -0.09774936 -0.06516624  
## 28351 0.076955 -0.1629156 -0.09774936 -0.06516624  
## 98576 0.075004 -0.1629156 -0.09774936 -0.06516624  
## 113468 0.068646 -0.1629156 -0.09774936 -0.06516624  
## 118468 0.074056 -0.1629120 -0.09775200 -0.06517200  
## price\_p1\_var\_complete price\_p2\_var\_complete price\_p3\_var\_complete  
## 23139 0.125976 0.103395 0.071536  
## 28351 0.131032 0.108896 0.076955  
## 98576 0.129444 0.106863 0.075004  
## 113468 0.123086 0.100505 0.068646  
## 118468 0.128132 0.105996 0.074056  
## price\_p1\_fix\_complete price\_p2\_fix\_complete price\_p3\_fix\_complete  
## 23139 -0.1629156 -0.09774936 -0.06516624  
## 28351 -0.1629156 -0.09774936 -0.06516624  
## 98576 -0.1629156 -0.09774936 -0.06516624  
## 113468 -0.1629156 -0.09774936 -0.06516624  
## 118468 -0.1629120 -0.09775200 -0.06517200

which(p\_1$price\_p2\_fix\_complete<0)

## [1] 23139 28351 98576 113468 118468 125820 128762 160828 181812

head(p\_1[c(23139, 28351, 98576, 113468, 118468, 125820, 128762, 160828, 181812), ],5L)

## id price\_date price\_p1\_var price\_p2\_var  
## 23139 951d99fe07ca94c2139f43bc37095139 2001-03-15 0.125976 0.103395  
## 28351 f7bdc6fa1067cd26fd80bfb9f3fca28f 2001-03-15 0.131032 0.108896  
## 98576 9b523ad5ba8aa2e524dcda5b3d54dab2 2001-02-15 0.129444 0.106863  
## 113468 cfd098ee6c567eb32374c77d20571bc7 2001-02-15 0.123086 0.100505  
## 118468 51d7d8a0bf6b8bd94f8c1de7942c66ea 2001-07-15 0.128132 0.105996  
## price\_p3\_var price\_p1\_fix price\_p2\_fix price\_p3\_fix  
## 23139 0.071536 -0.1629156 -0.09774936 -0.06516624  
## 28351 0.076955 -0.1629156 -0.09774936 -0.06516624  
## 98576 0.075004 -0.1629156 -0.09774936 -0.06516624  
## 113468 0.068646 -0.1629156 -0.09774936 -0.06516624  
## 118468 0.074056 -0.1629120 -0.09775200 -0.06517200  
## price\_p1\_var\_complete price\_p2\_var\_complete price\_p3\_var\_complete  
## 23139 0.125976 0.103395 0.071536  
## 28351 0.131032 0.108896 0.076955  
## 98576 0.129444 0.106863 0.075004  
## 113468 0.123086 0.100505 0.068646  
## 118468 0.128132 0.105996 0.074056  
## price\_p1\_fix\_complete price\_p2\_fix\_complete price\_p3\_fix\_complete  
## 23139 -0.1629156 -0.09774936 -0.06516624  
## 28351 -0.1629156 -0.09774936 -0.06516624  
## 98576 -0.1629156 -0.09774936 -0.06516624  
## 113468 -0.1629156 -0.09774936 -0.06516624  
## 118468 -0.1629120 -0.09775200 -0.06517200

which(p\_1$price\_p3\_fix\_complete<0)

## [1] 23139 28351 98576 113468 118468 125820 128762 160828 181812

head(p\_1[c( 23139, 28351, 98576, 113468, 118468, 125820, 128762, 160828, 181812), ],5L)

## id price\_date price\_p1\_var price\_p2\_var  
## 23139 951d99fe07ca94c2139f43bc37095139 2001-03-15 0.125976 0.103395  
## 28351 f7bdc6fa1067cd26fd80bfb9f3fca28f 2001-03-15 0.131032 0.108896  
## 98576 9b523ad5ba8aa2e524dcda5b3d54dab2 2001-02-15 0.129444 0.106863  
## 113468 cfd098ee6c567eb32374c77d20571bc7 2001-02-15 0.123086 0.100505  
## 118468 51d7d8a0bf6b8bd94f8c1de7942c66ea 2001-07-15 0.128132 0.105996  
## price\_p3\_var price\_p1\_fix price\_p2\_fix price\_p3\_fix  
## 23139 0.071536 -0.1629156 -0.09774936 -0.06516624  
## 28351 0.076955 -0.1629156 -0.09774936 -0.06516624  
## 98576 0.075004 -0.1629156 -0.09774936 -0.06516624  
## 113468 0.068646 -0.1629156 -0.09774936 -0.06516624  
## 118468 0.074056 -0.1629120 -0.09775200 -0.06517200  
## price\_p1\_var\_complete price\_p2\_var\_complete price\_p3\_var\_complete  
## 23139 0.125976 0.103395 0.071536  
## 28351 0.131032 0.108896 0.076955  
## 98576 0.129444 0.106863 0.075004  
## 113468 0.123086 0.100505 0.068646  
## 118468 0.128132 0.105996 0.074056  
## price\_p1\_fix\_complete price\_p2\_fix\_complete price\_p3\_fix\_complete  
## 23139 -0.1629156 -0.09774936 -0.06516624  
## 28351 -0.1629156 -0.09774936 -0.06516624  
## 98576 -0.1629156 -0.09774936 -0.06516624  
## 113468 -0.1629156 -0.09774936 -0.06516624  
## 118468 -0.1629120 -0.09775200 -0.06517200

## Replacing negative values with median to maintain the data structure

p\_1$price\_p1\_fix\_complete = replace(p\_1$price\_p1\_fix\_complete, p\_1$price\_p1\_fix\_complete<0, median(p\_1$price\_p1\_fix\_complete))  
   
p\_1$price\_p2\_fix\_complete = replace(p\_1$price\_p2\_fix\_complete, p\_1$price\_p2\_fix\_complete<0, median(p\_1$price\_p2\_fix\_complete))  
   
p\_1$price\_p3\_fix\_complete = replace(p\_1$price\_p3\_fix\_complete, p\_1$price\_p3\_fix\_complete<0, median(p\_1$price\_p3\_fix\_complete))

## Checking if any any negative values still exist

which(p\_1$price\_p1\_fix\_complete<0)

## integer(0)

which(p\_1$price\_p2\_fix\_complete<0)

## integer(0)

which(p\_1$price\_p3\_var\_complete<0)

## integer(0)