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)
```

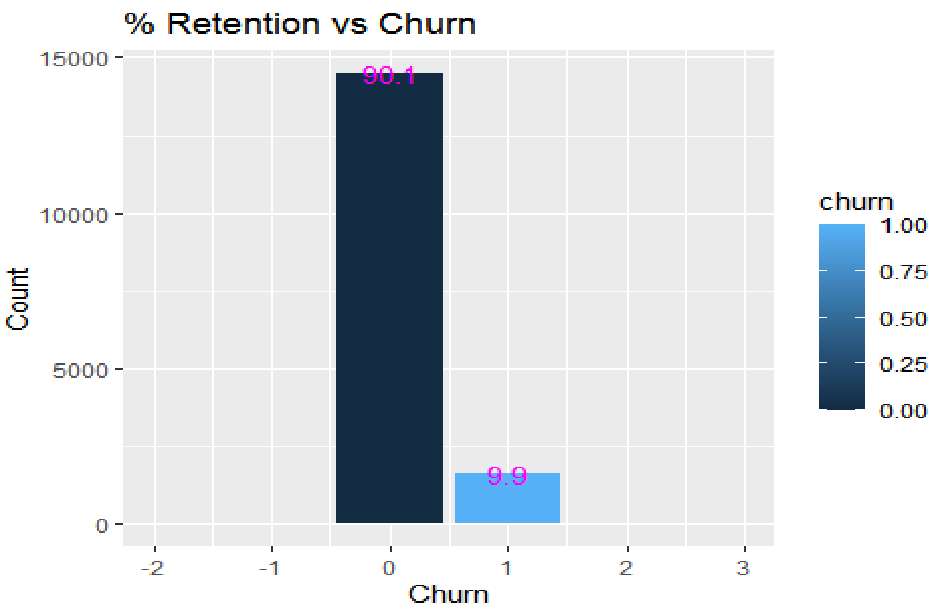
```
##                       id             activity_new        campaign_disc_ele
##               0.00000000               0.00000000             100.00000000
##            channel_sales                 cons_12m             cons_gas_12m
##               0.00000000               0.00000000               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")
```

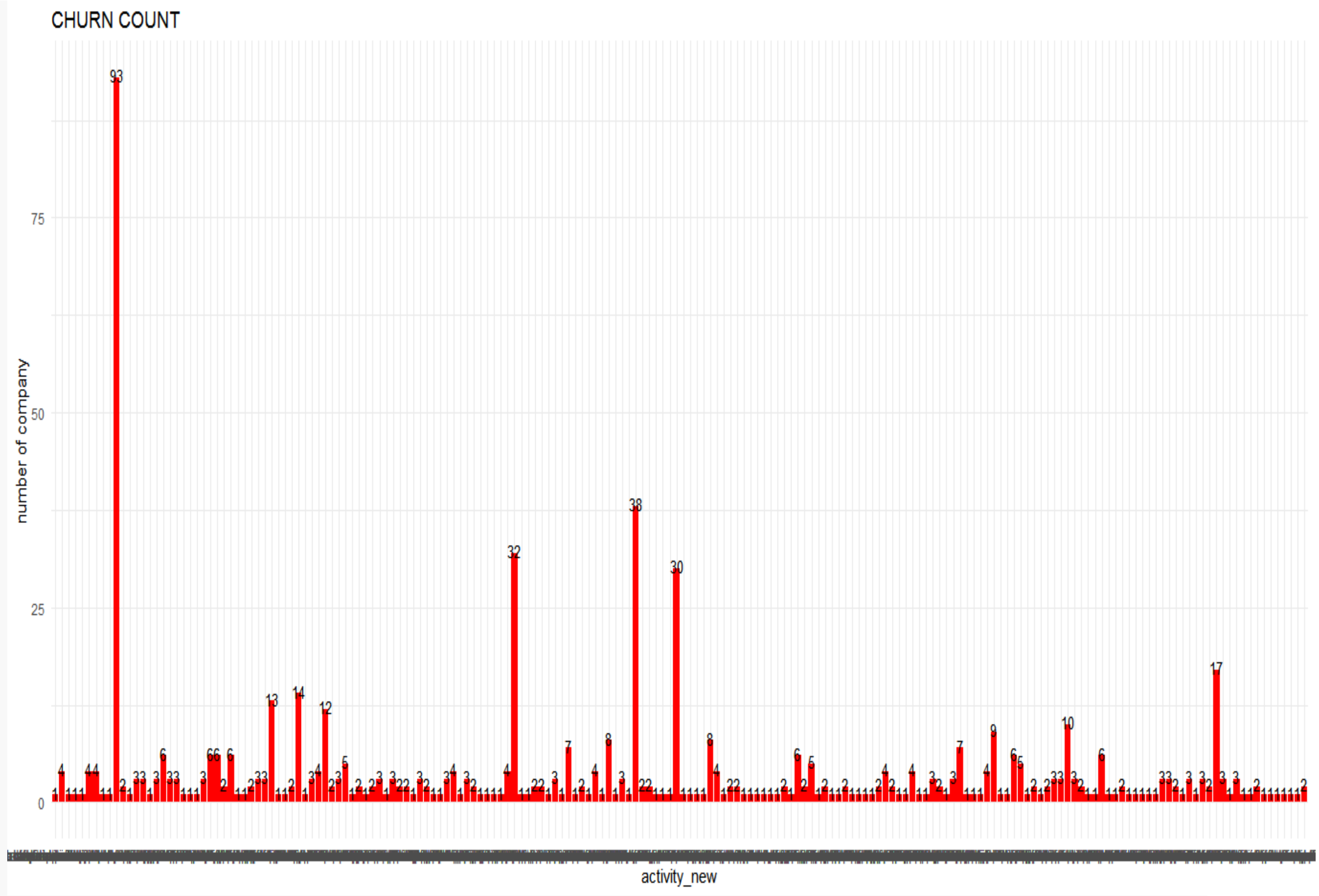
activity dataset with % Churn and % Retention

```
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
```

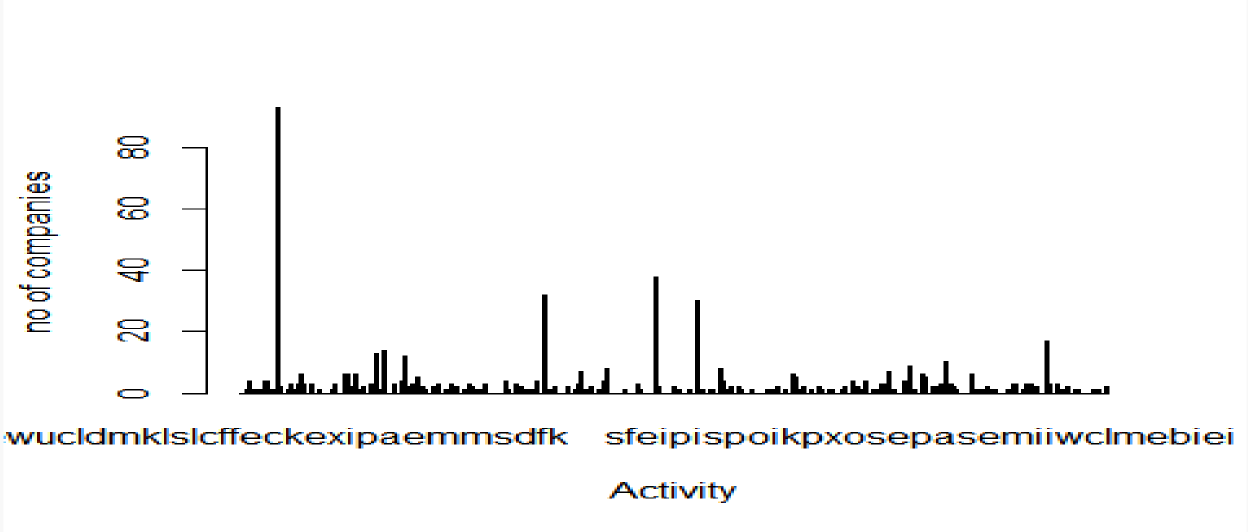
Visualization for Churn

```
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()
```



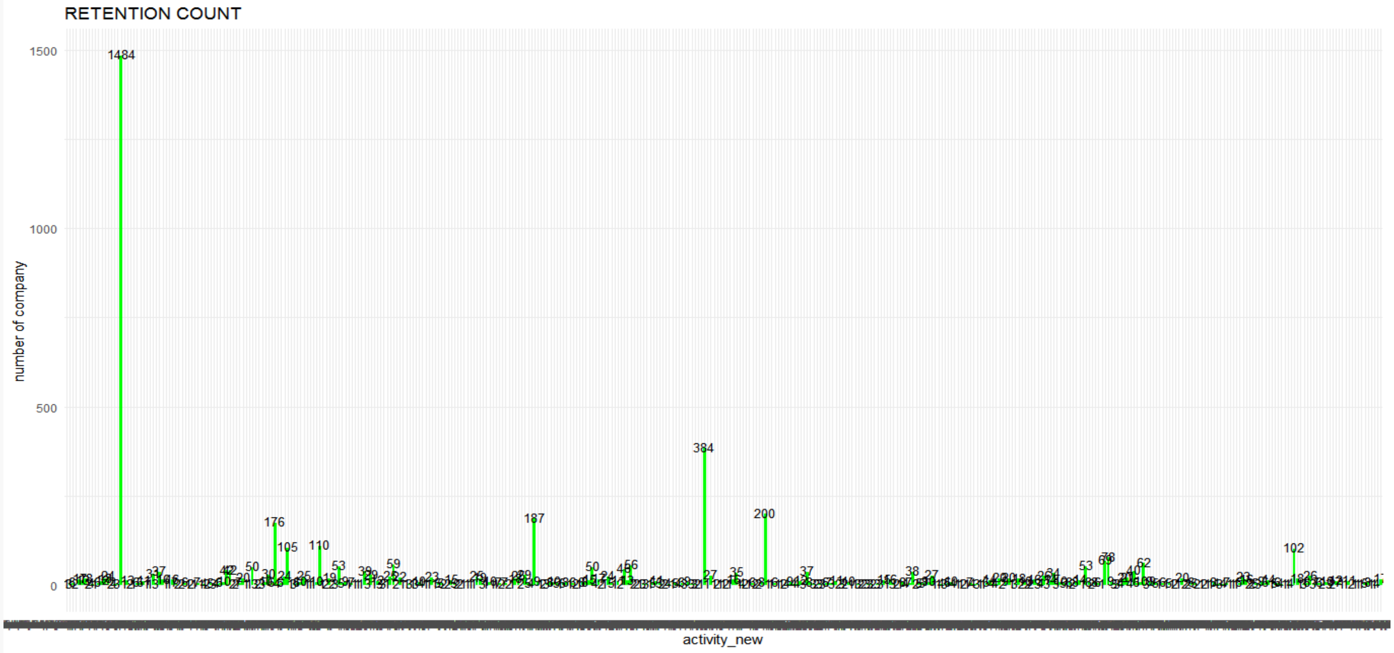
OR

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



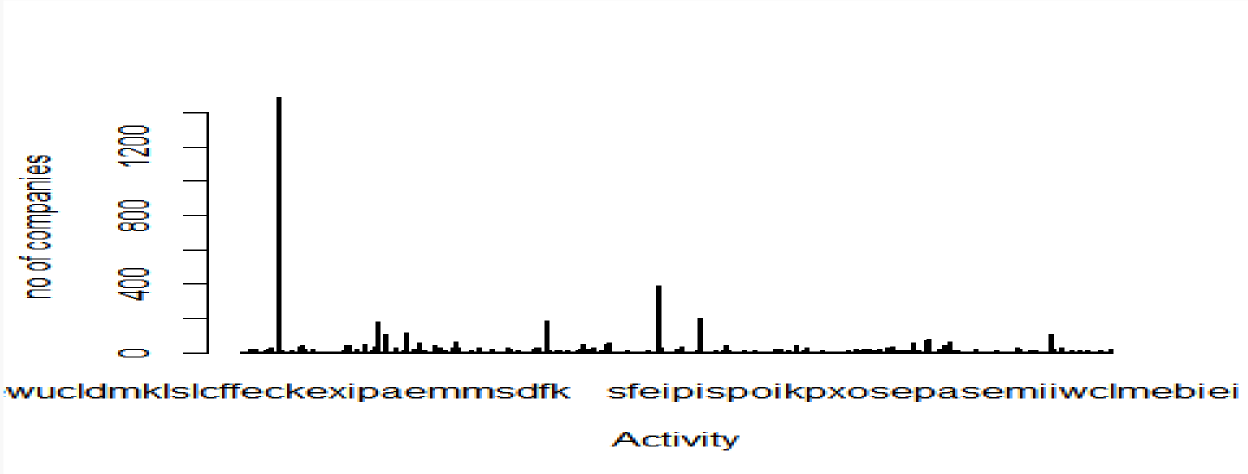
Visualization for Retention

```
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")
```

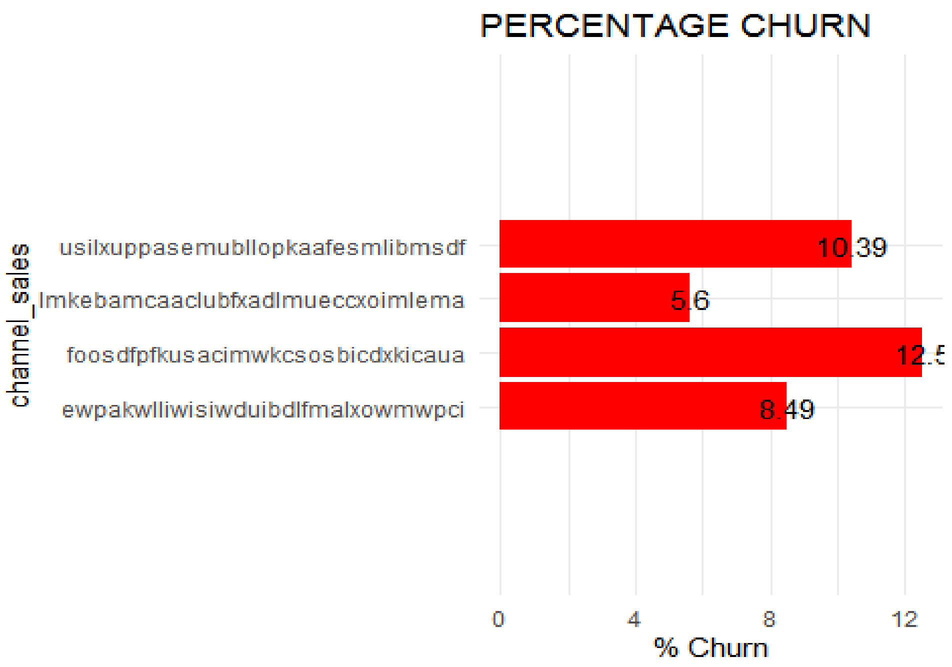
Channel_Sales dataset with % Churn and % Retention is REQUIRED

```
 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

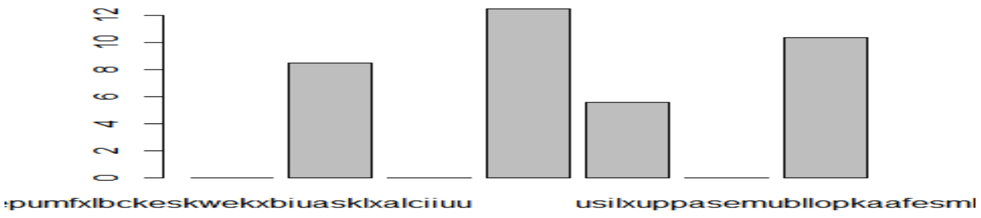
Visualization for Churn

```
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
```



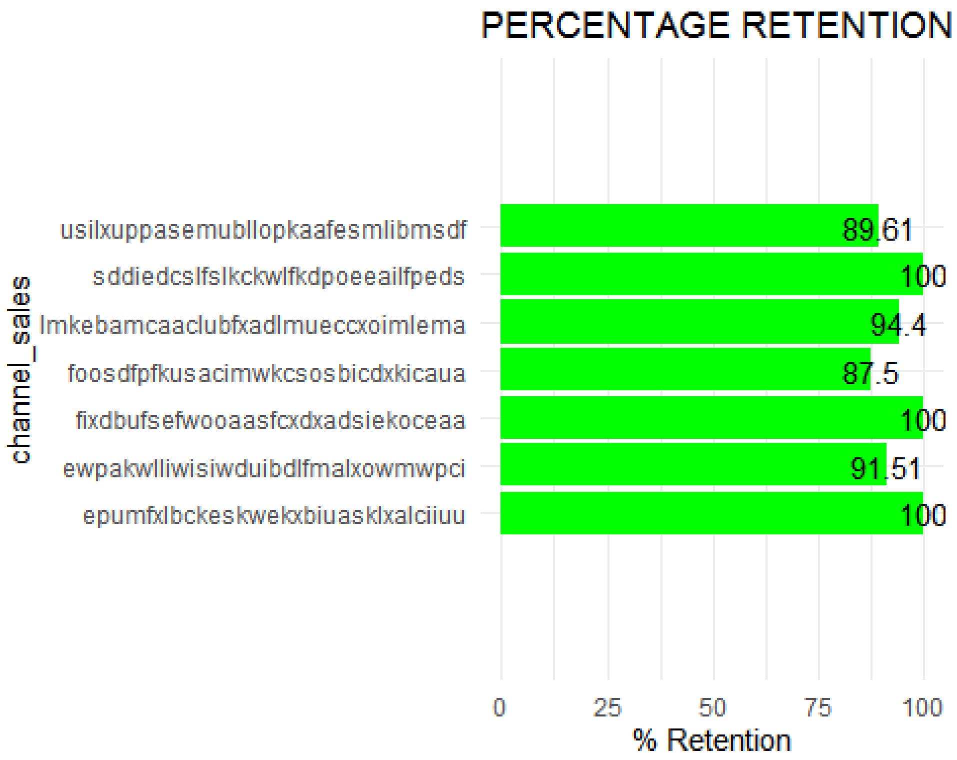
OR

```r
barplot(sales_2$Percent_churn, names.arg = sales_2$channel_sales)
```



Visualization for Retention

```r
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
```



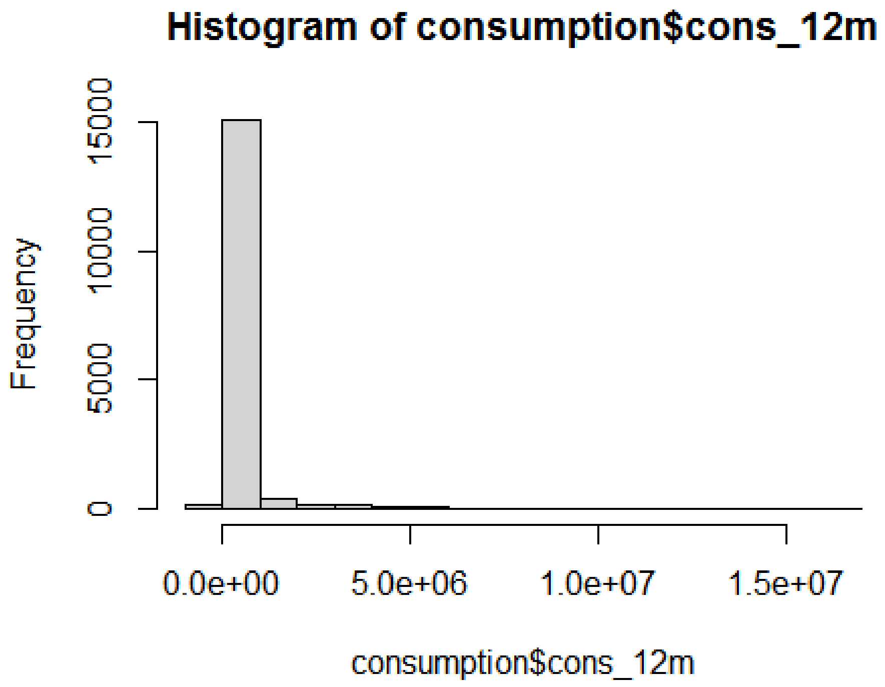
Consumption Distribution

```
consumption= train %>%
  select(id,cons_12m,cons_gas_12m,cons_last_month,imp_cons,has_gas,churn)
```

Histogram for cons_12m

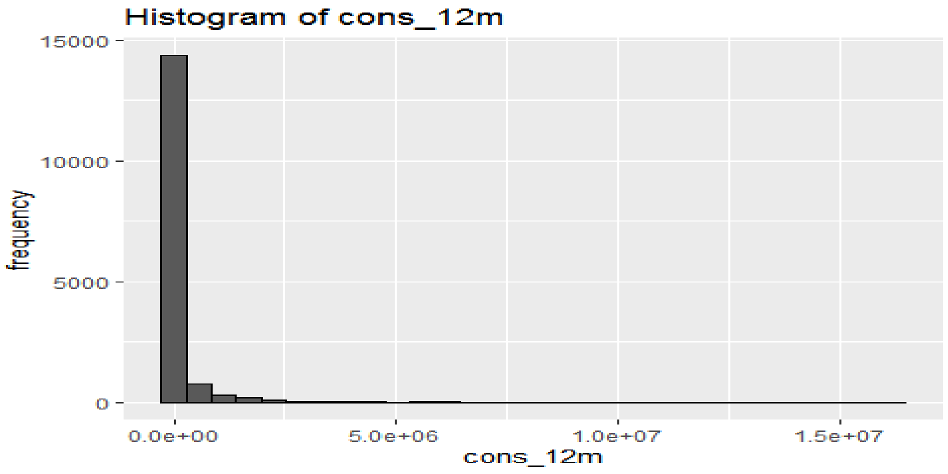
For Total frequency distribution (Histogram) i.e churn + retention

```
hist(consumption$cons_12m)
```



OR

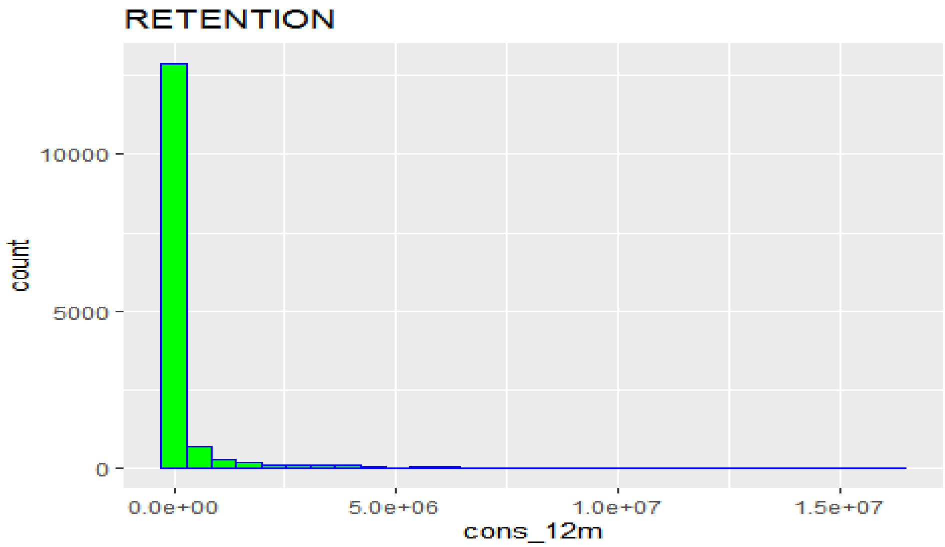
```
qplot(consumption$cons_12m, geom = "histogram",
      colour=I("black"),
      xlab = "cons_12m",
      ylab = "frequency",
      main = "Histogram of cons_12m")
```



To calculate churn and retention separately

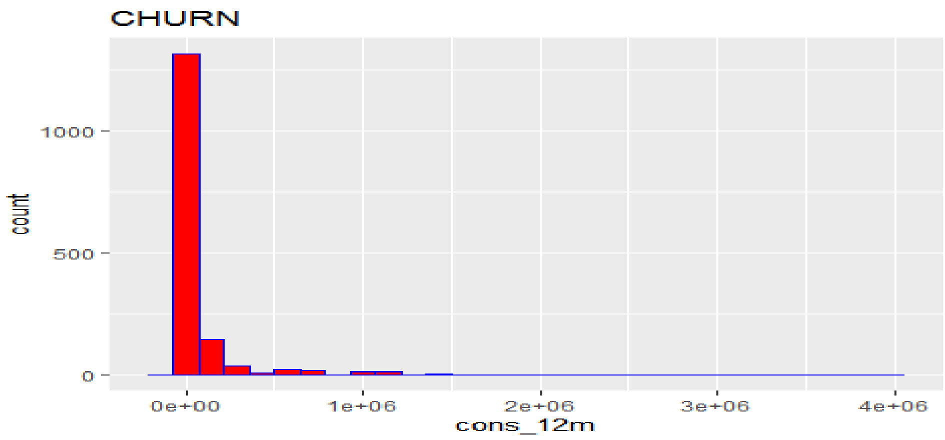
Histogram for cons_12m (RETENTION)

```
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)

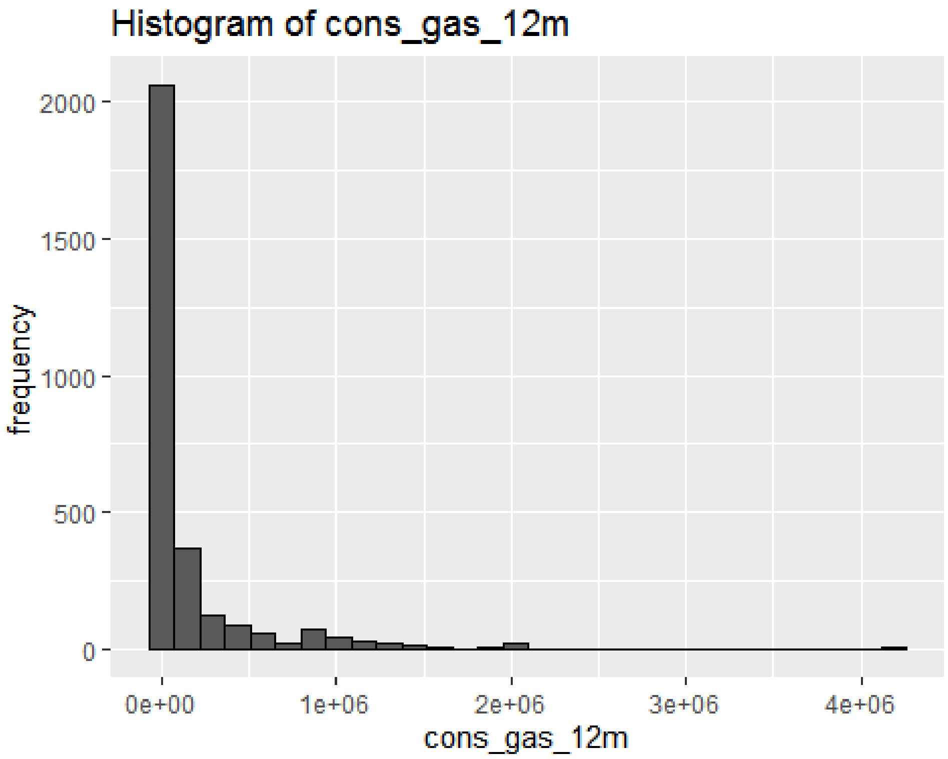
```
 consumption %>% filter(churn==1) %>%
  ggplot(aes(cons_12m)) +
  geom_histogram(fill="red",color = I("blue")) +
  ggtitle("CHURN")

## `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

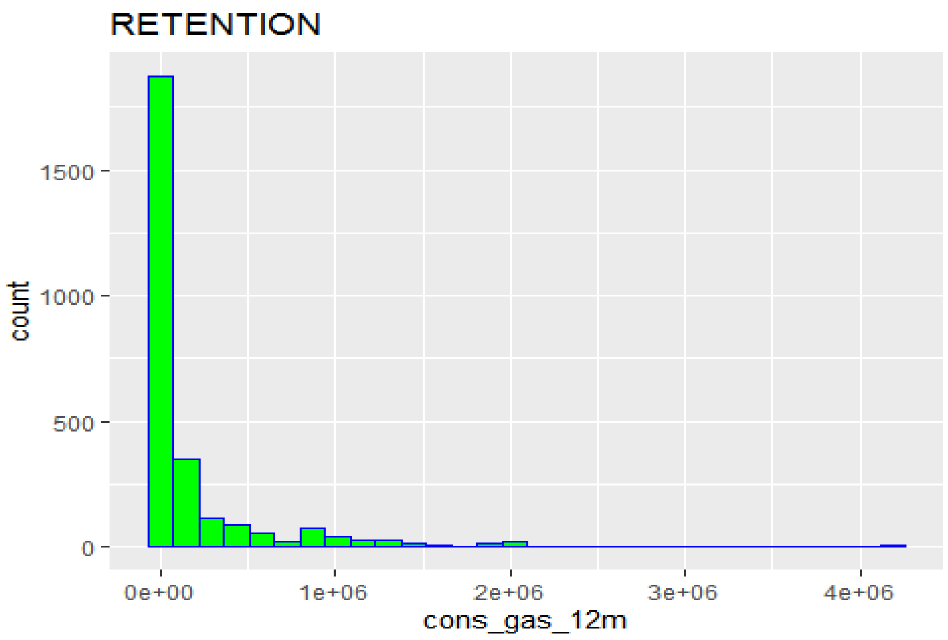
```
t_gas=consumption %>% filter(has_gas=="t") %>% select(cons_gas_12m,has_gas)

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

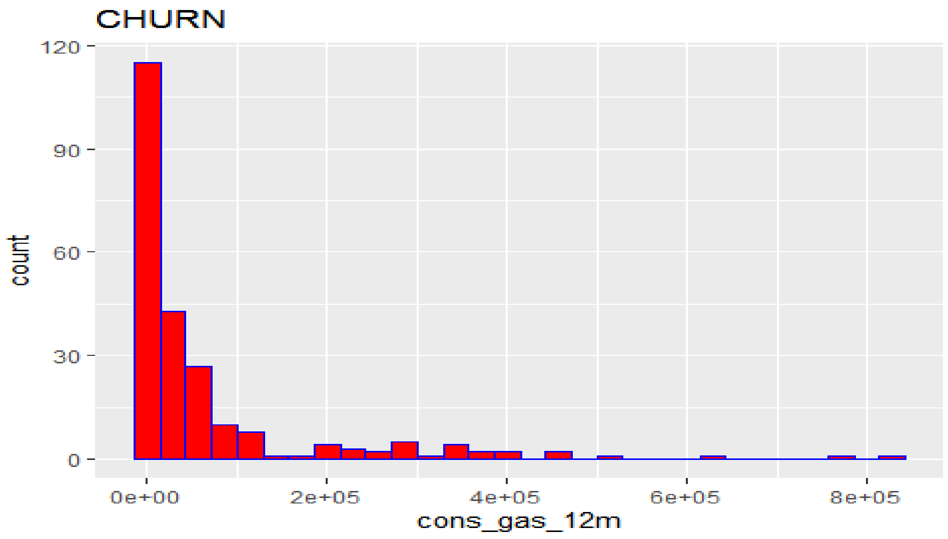
Histogram for cons_gas_12m (RETENTION)

```
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)

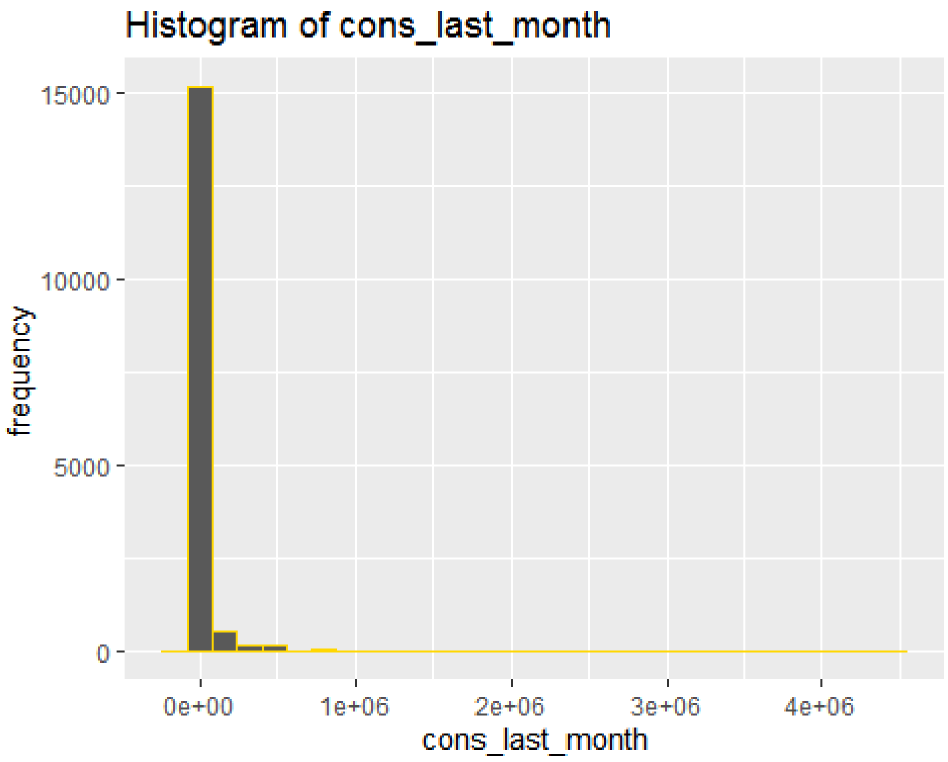
```
consumption %>% filter(has_gas=="t",churn==1) %>%
  ggplot(aes(cons_gas_12m)) +
  geom_histogram(fill="red",color = I("blue")) +
  ggtitle("CHURN")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Histogram for cons_last_month

For Total (churn + retention) Frequency/Count Distribution

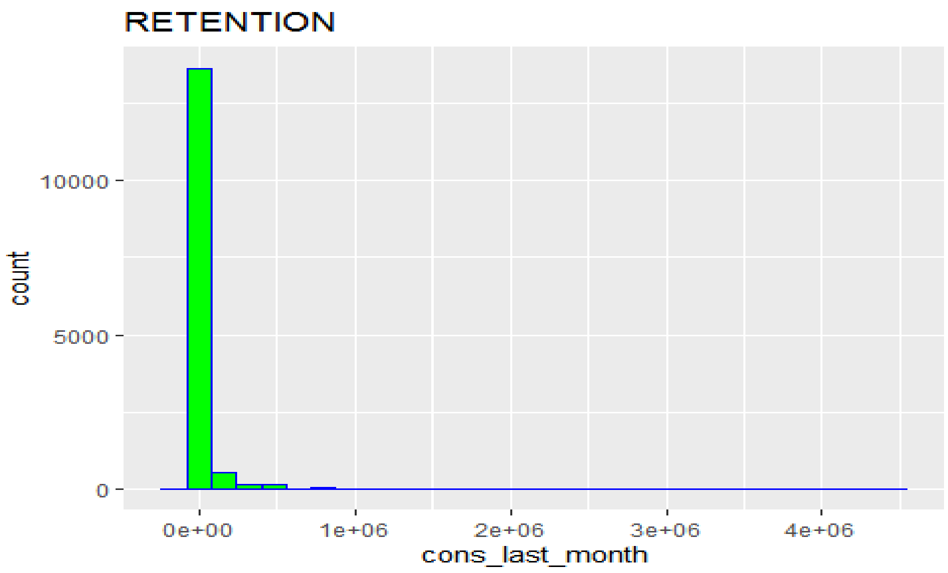
```
qplot(consumption$cons_last_month, geom = "histogram",
      color = I("GOLD"),
      xlab = "cons_last_month",
      ylab = "frequency",
      main = "Histogram of cons_last_month")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



To calculate churn and retention separately

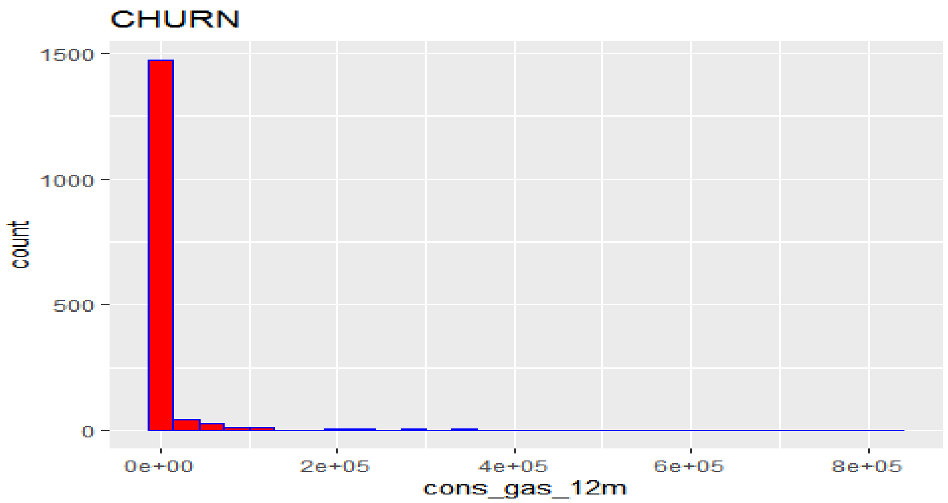
Histogram for cons_last_month (RETENTION)

```
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)

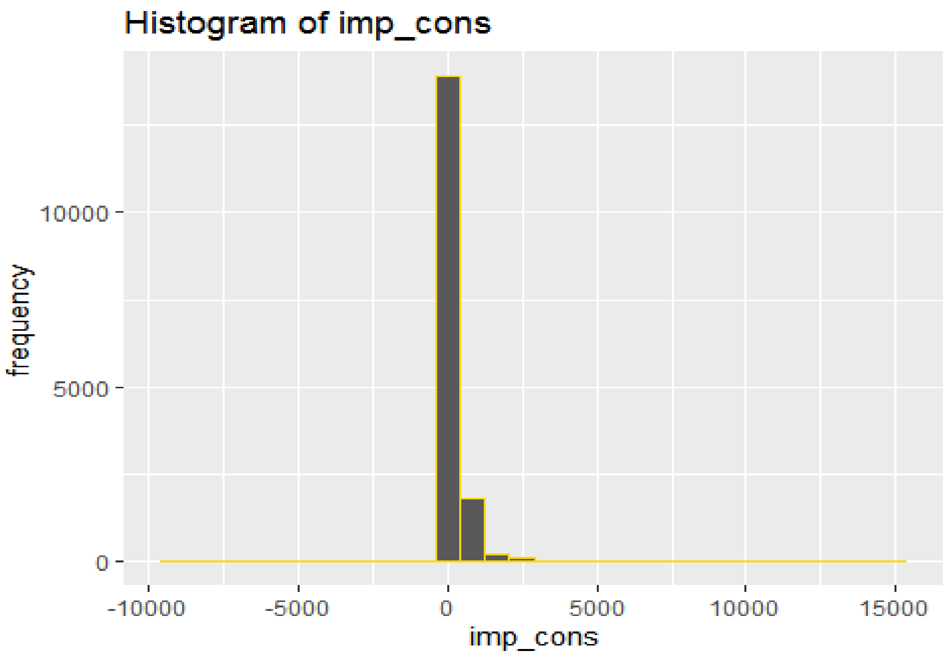
```
consumption %>% filter(churn==1) %>%
  ggplot(aes(cons_gas_12m)) +
  geom_histogram(fill="red",color = I("blue")) +
  ggtitle("CHURN")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Histogram for imp_cons

For Total (churn + retention) Frequency/Count Distribution

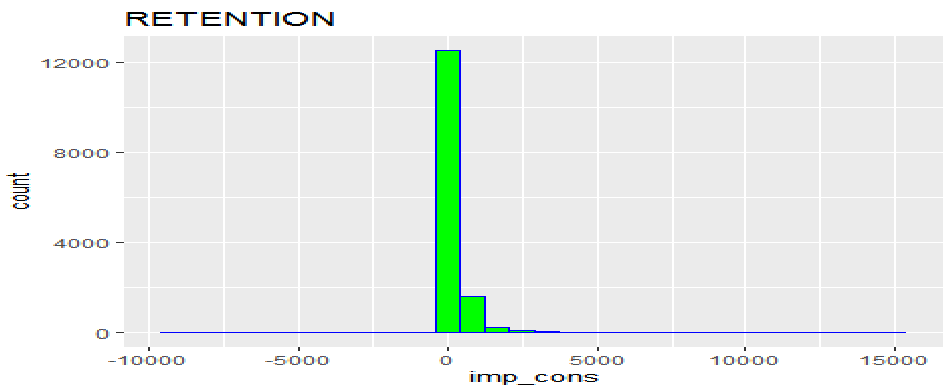
```
qplot(consumption$imp_cons, geom = "histogram", # cld input binwidth=40 to
get more insights
      color = I("GOLD"),
      xlab = "imp_cons",
      ylab = "frequency",
      main = "Histogram of imp_cons")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



To calculate churn and retention separately

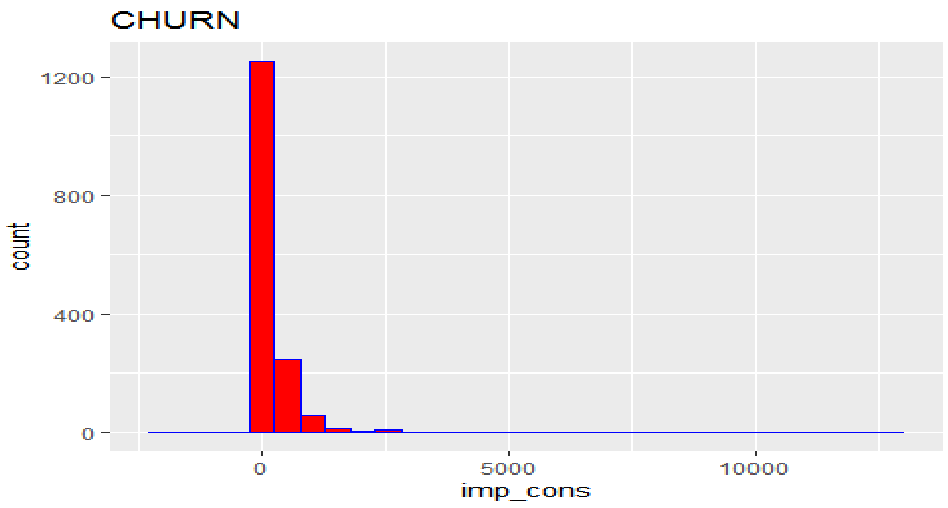
Histogram for imp_cons (RETENTION)

```
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`.
```



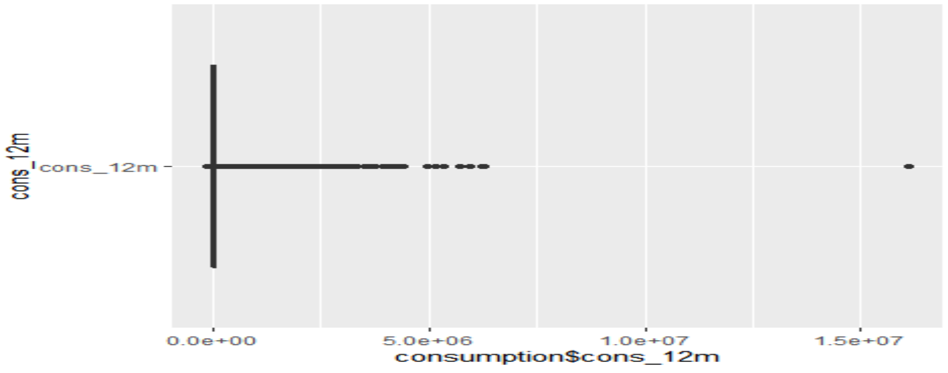
Histogram for imp_cons (CHURN)

```
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`.
```

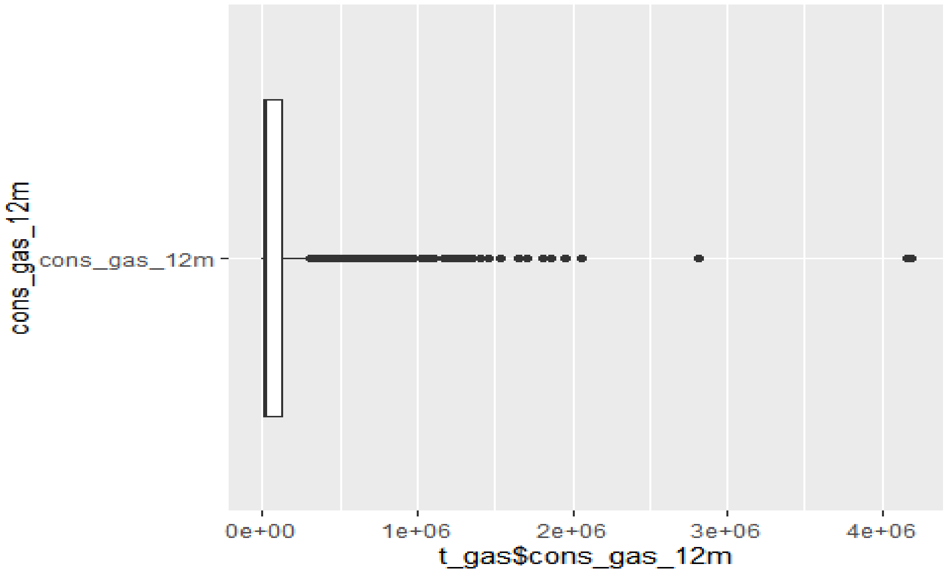


Box Plots

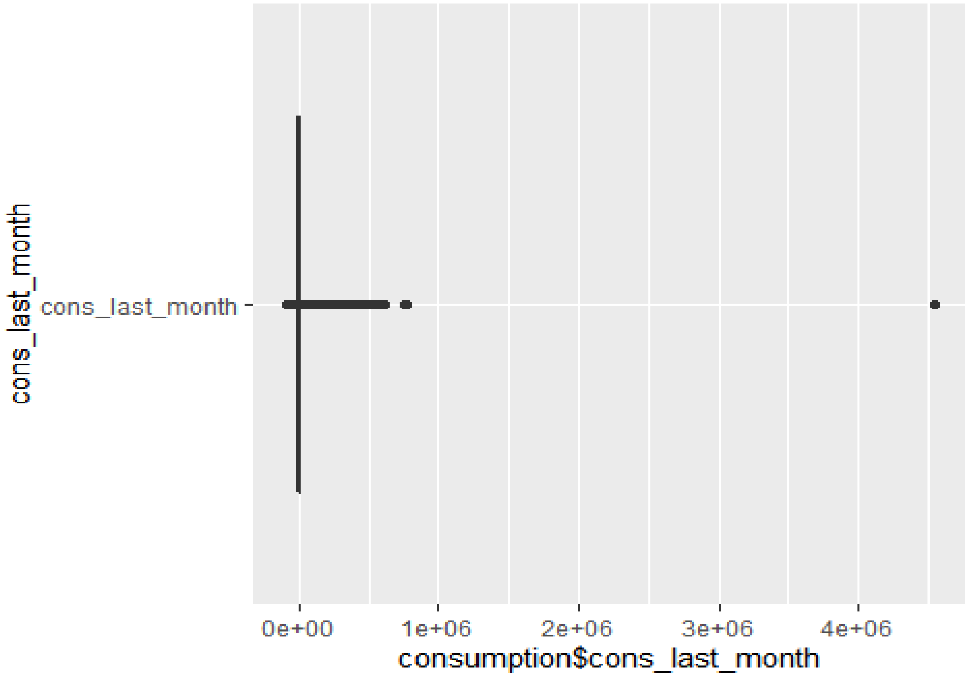
```
 qplot("cons_12m", consumption$cons_12m, geom = "boxplot") + coord_flip()
```



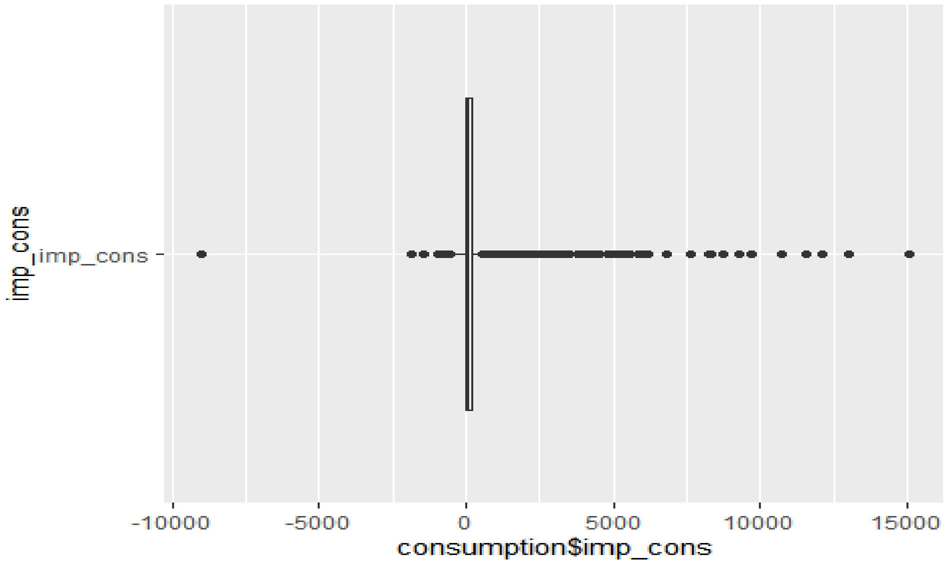
```
qplot("cons_gas_12m", t_gas$cons_gas_12m, geom = "boxplot") + coord_flip()
```



```
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
```

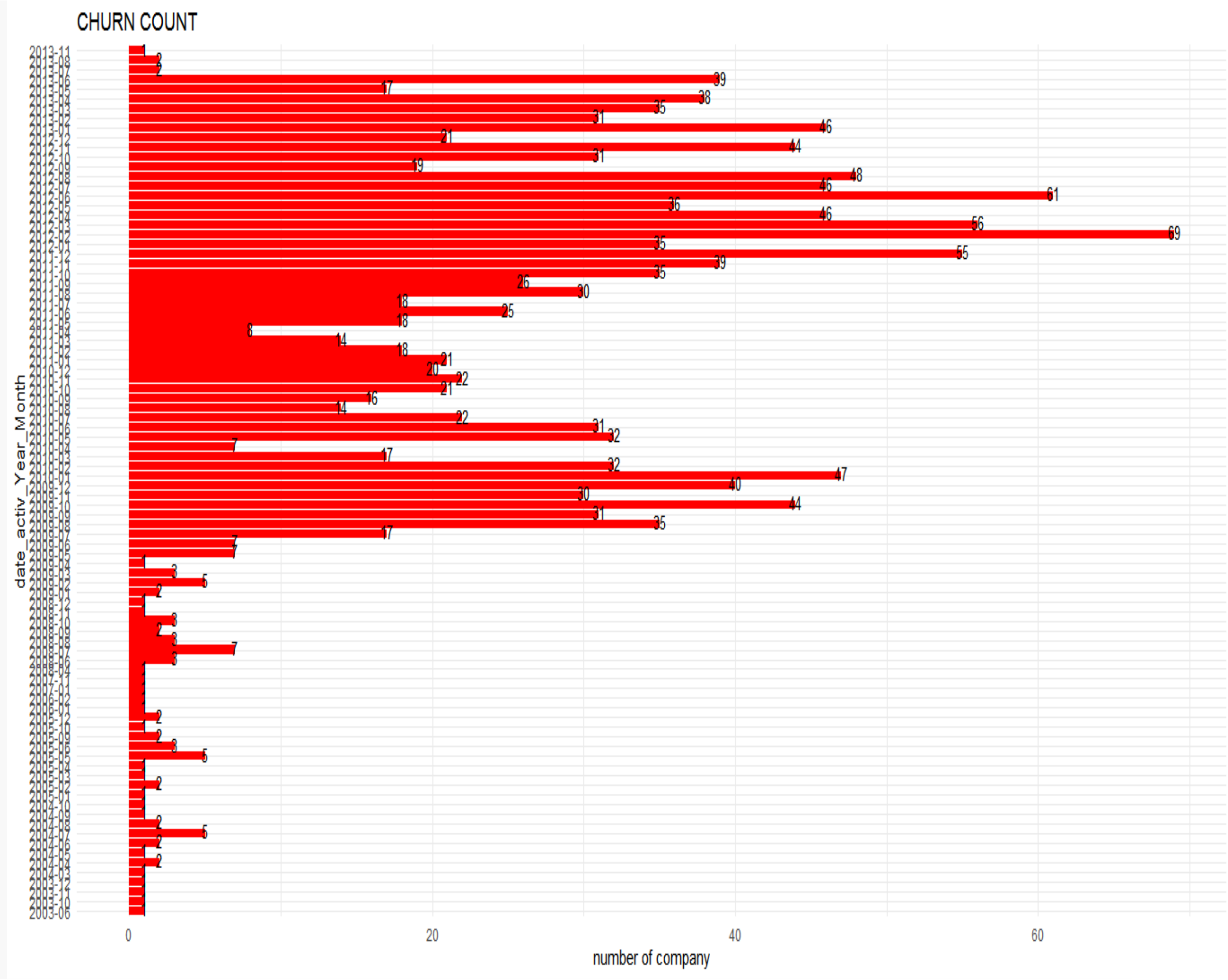
```
class(d1)

## [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
```

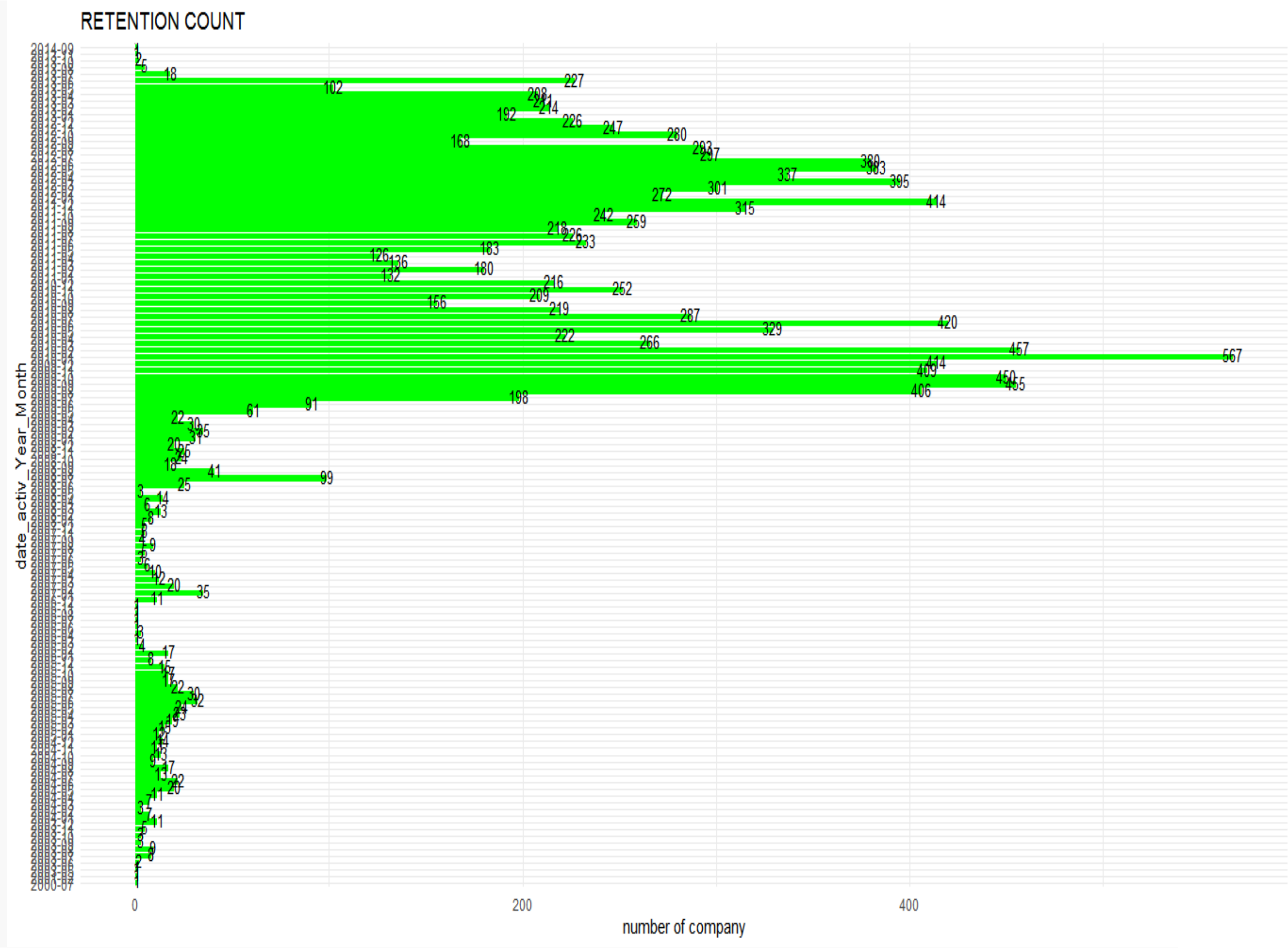
Visualization for Churn

```
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"
```

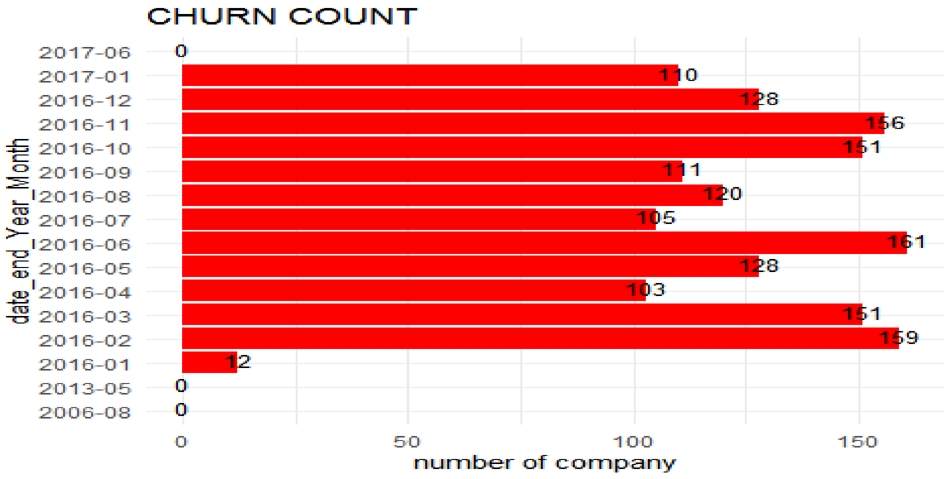
```
d2 = as.data.frame(d2) ## RATE LIMITING STEP; VERY IMPORTANT
colnames(d2) = c("date_end_Year_Month","retention","churn")
```

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
```

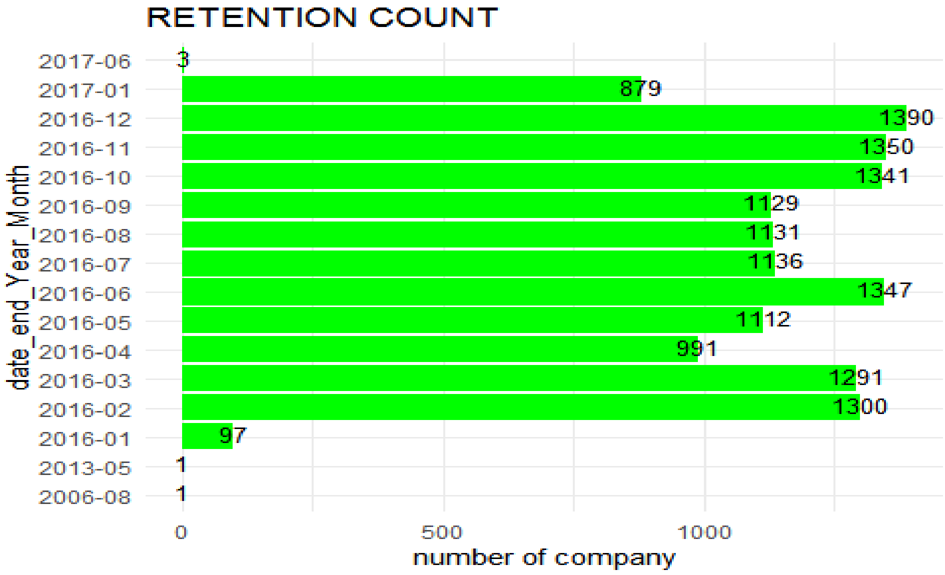
Visualization for Churn

```
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"
```

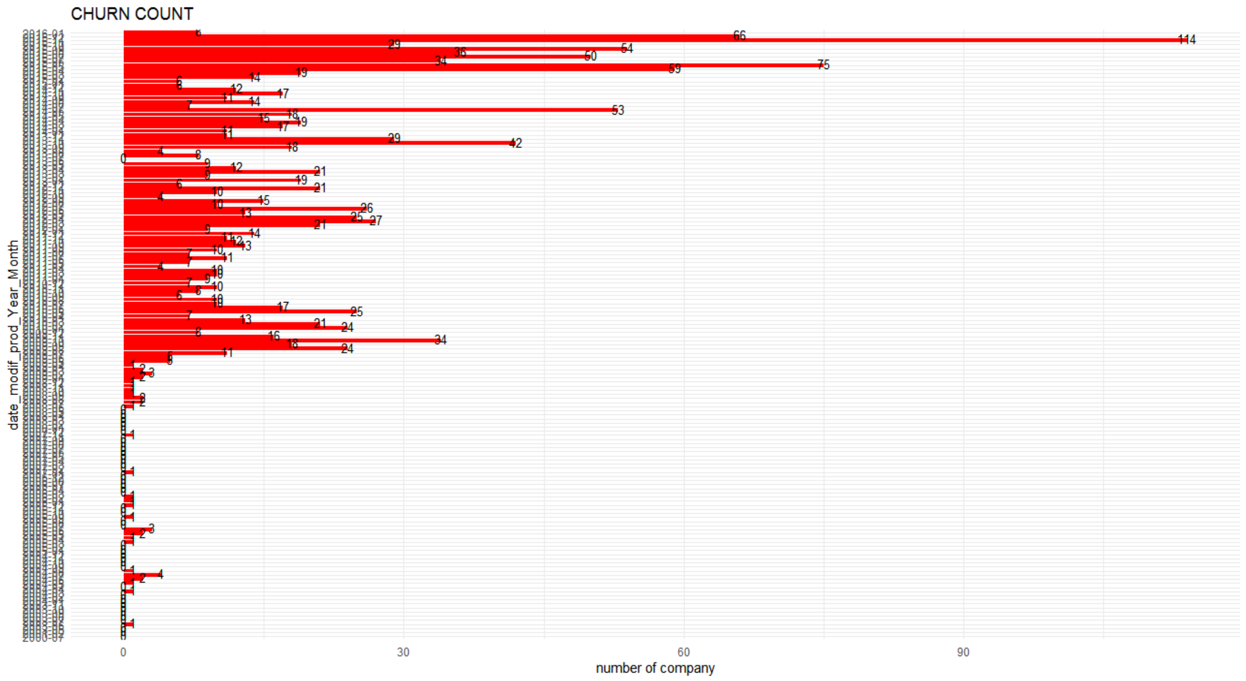
```
d3 = as.data.frame(d3) ## RATE LIMITING STEP; VERY IMPORTANT
colnames(d3) = c("date_modif_prod_Year_Month","retention","churn")
```

```
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
```

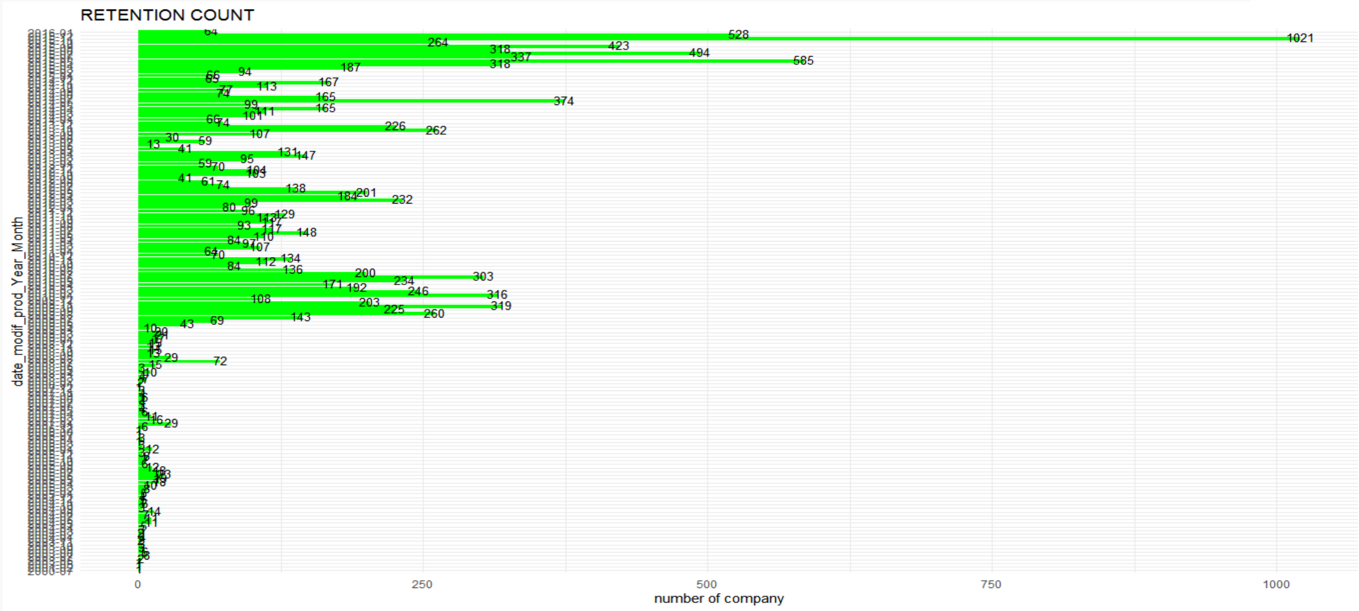
Visualization for Churn

```
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

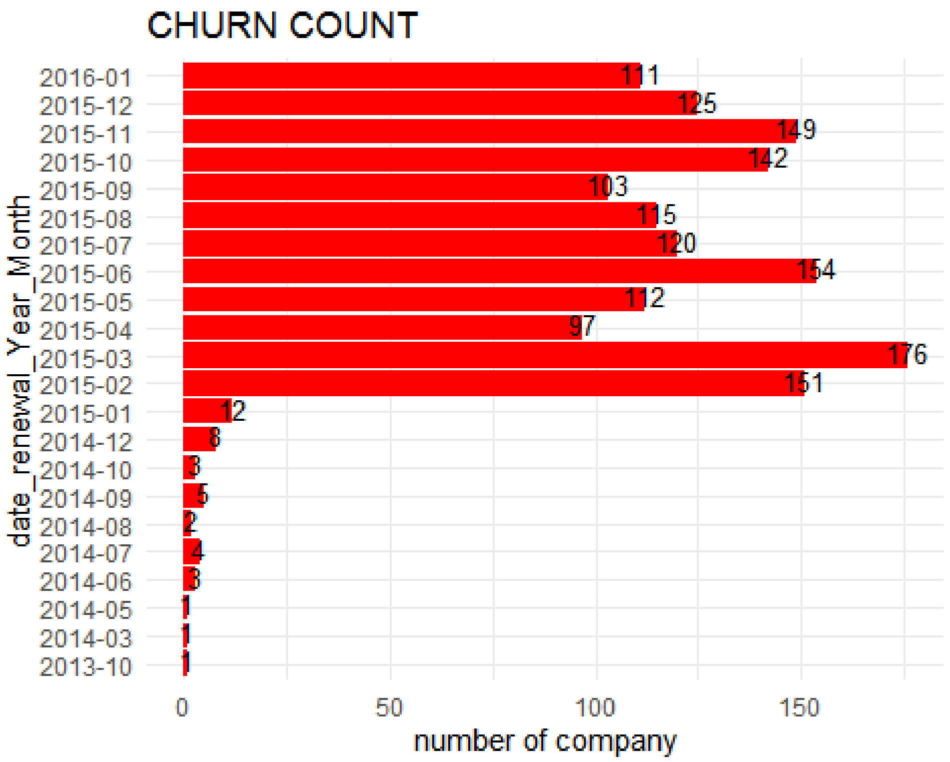
```
d4 = dates %>%
   group_by(date_renewal_Year_Month,churn,id) %>%
   select(date_renewal_Year_Month,churn,id) %>%
   summarise(n=n()) %>%
   summarise(n=n()) %>%
   spread("churn", "n")

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
```

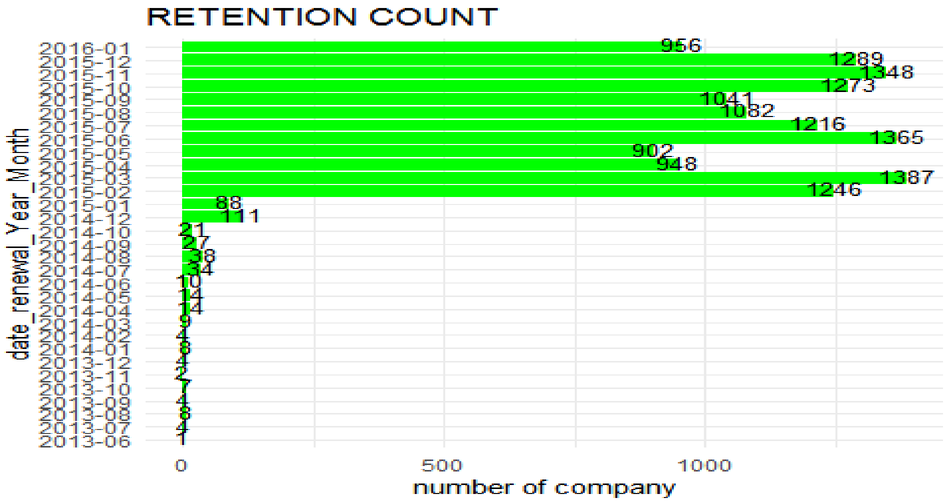
Visualization for Churn

```
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)
```

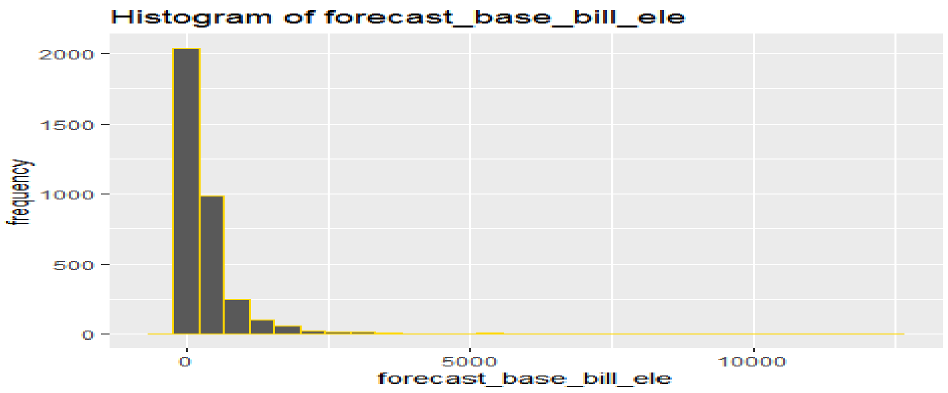
```
##  [1] "id"                       "forecast_base_bill_ele"  
##  [3] "forecast_base_bill_year"  "forecast_bill_12m"       
##  [5] "forecast_cons"            "forecast_cons_12m"       
##  [7] "forecast_cons_year"       "forecast_discount_energy"
##  [9] "forecast_meter_rent_12m"  "forecast_price_energy_p1"
## [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
```

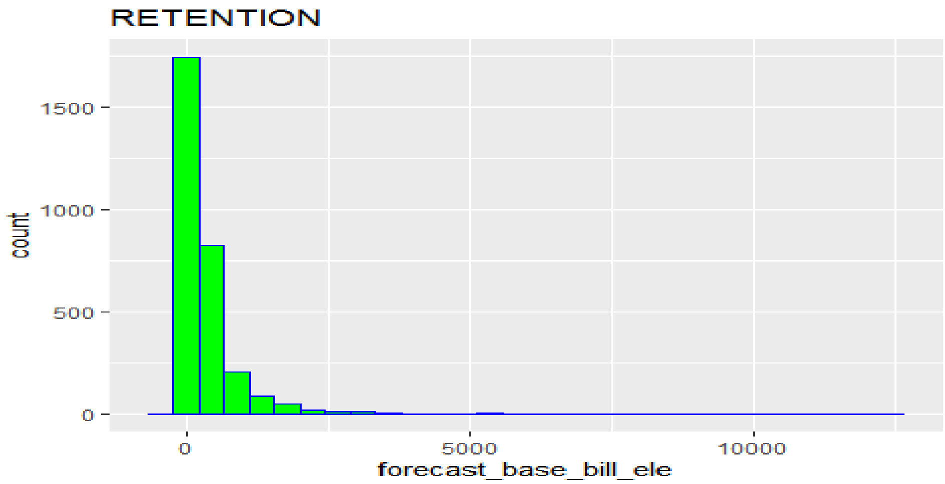
Total (retention + churn)

```
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")
```



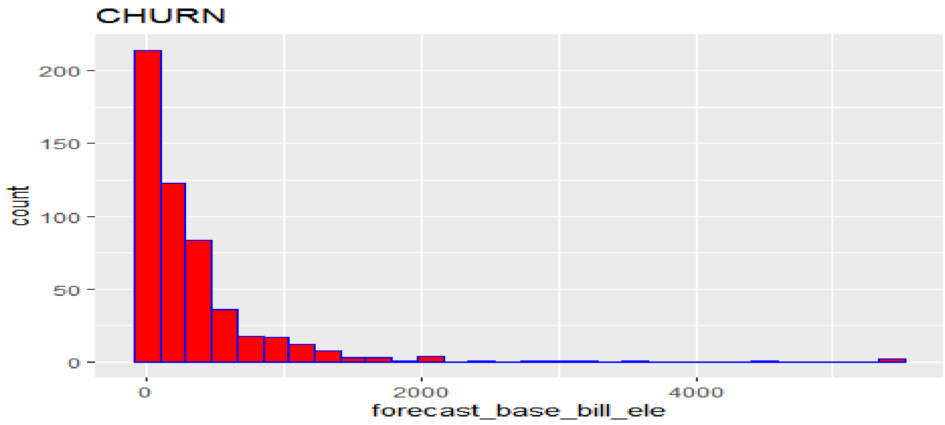
Histogram for forecast_base_bill_ele (RETENTION)

```
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).
```



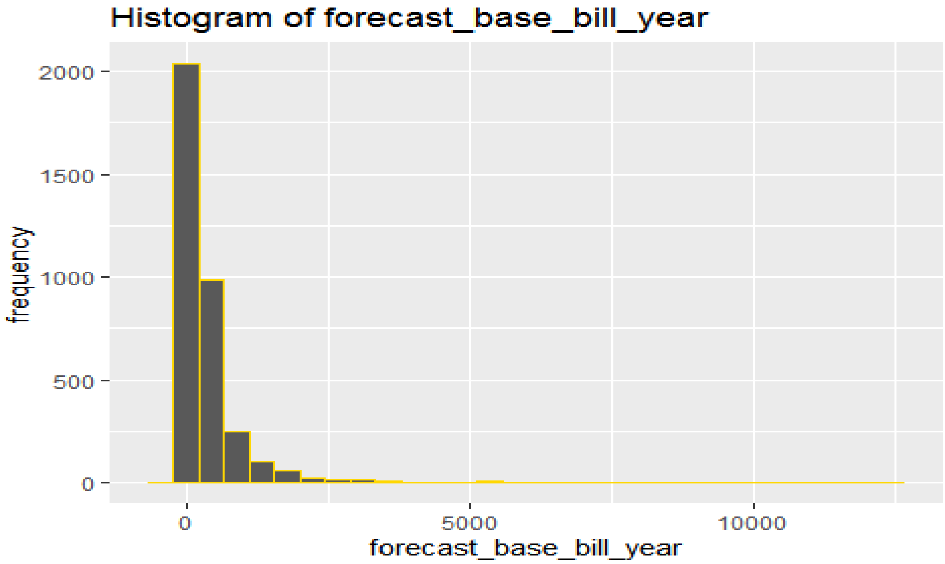
Histogram for forecast_base_bill_ele (CHURN)

```
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).
```



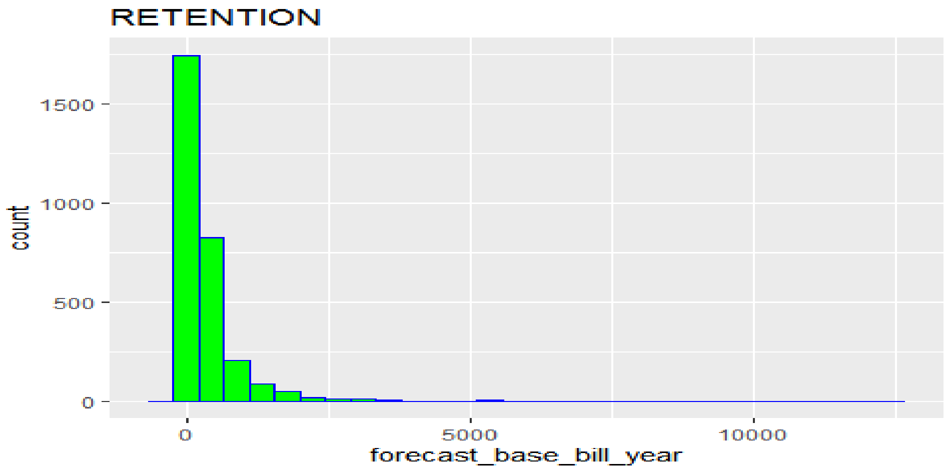
Total (retention + churn)

```r
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)

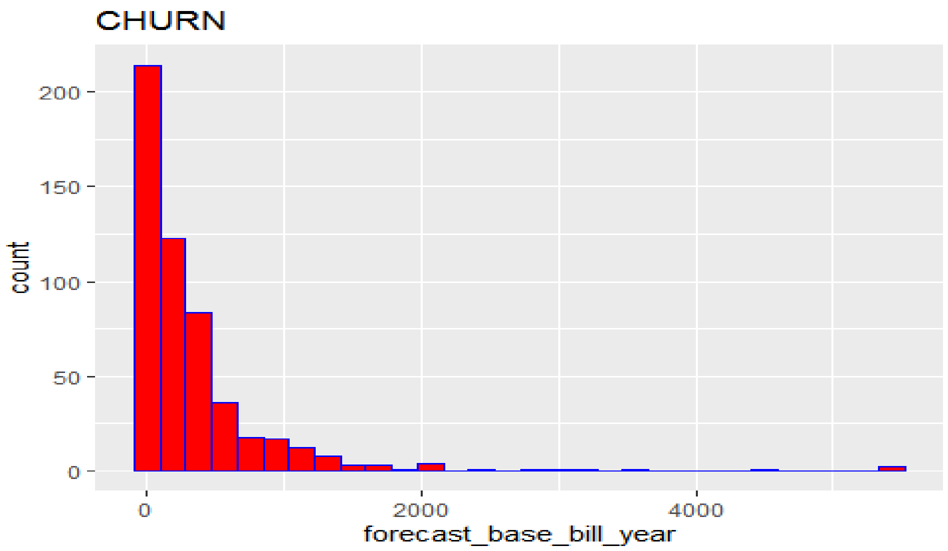
```r
  forecast %>%
  filter(churn==0) %>%
  ggplot(aes(forecast_base_bill_year)) +
  geom_histogram(fill="green",color = I("blue")) +
  ggtitle("RETENTION")
```



Histogram for forecast_base_bill_year (CHURN)

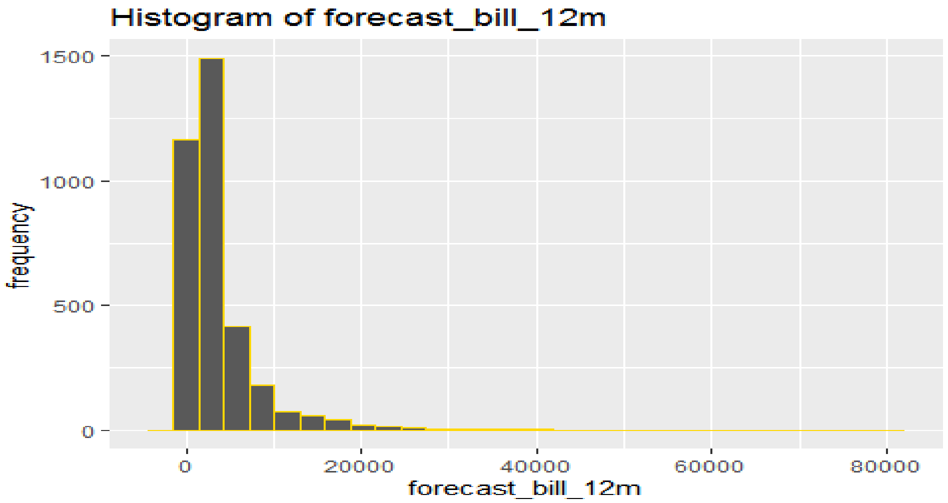
```
   forecast %>% filter(churn==1) %>%
   ggplot(aes(forecast_base_bill_year)) +
   geom_histogram(fill="red",color = I("blue")) +
   ggtitle("CHURN")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1064 rows containing non-finite values (stat_bin).
```



Total (retention + churn)

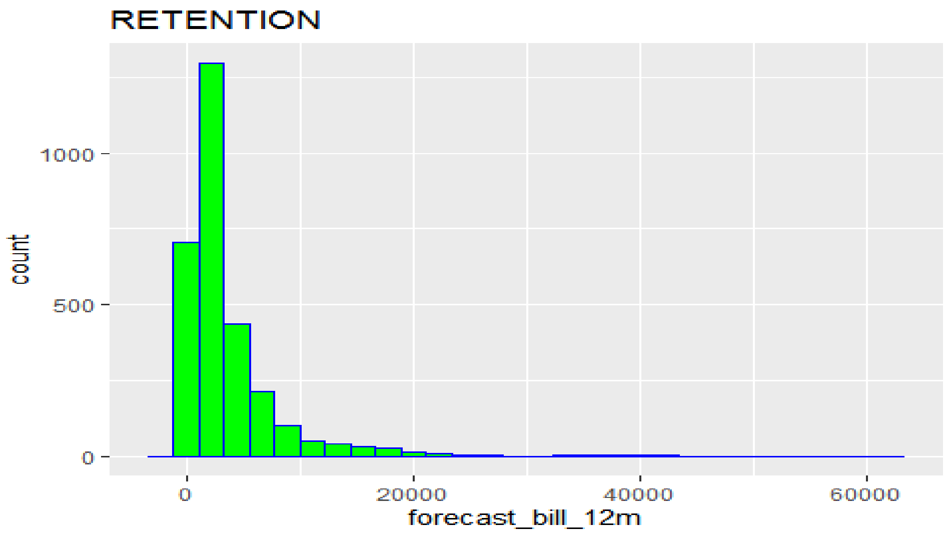
```
qplot(forecast$forecast_bill_12m, geom = "histogram",
       color = I("GOLD"),
       xlab = "forecast_bill_12m",
       ylab = "frequency",
       main = "Histogram of forecast_bill_12m")
```



Histogram for forecast_bill_12m (RETENTION)

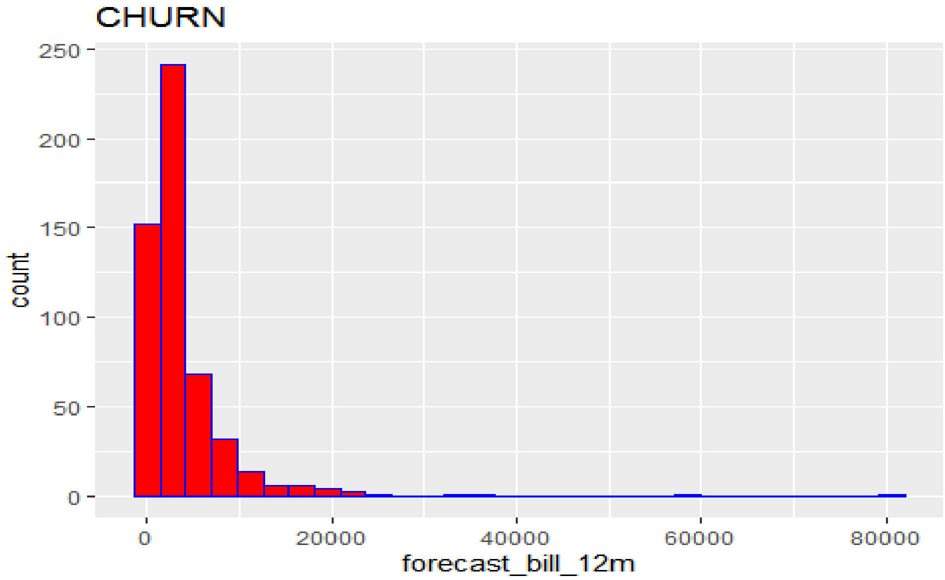
```
 forecast %>% filter(churn==0) %>%
   ggplot(aes(forecast_bill_12m)) +
   geom_histogram(fill="green",color = I("blue")) +
   ggtitle("RETENTION")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 11524 rows containing non-finite values (stat_bin).
```



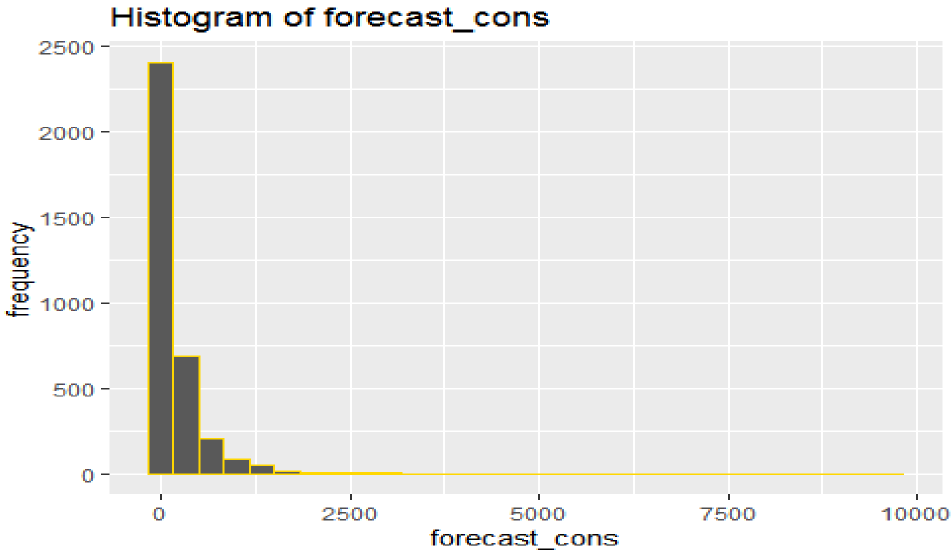
Histogram for forecast_bill_12m (CHURN)

```
forecast %>% filter(churn==1) %>%
   ggplot(aes(forecast_bill_12m)) +
   geom_histogram(fill="red",color = I("blue")) +
   ggtitle("CHURN")
```



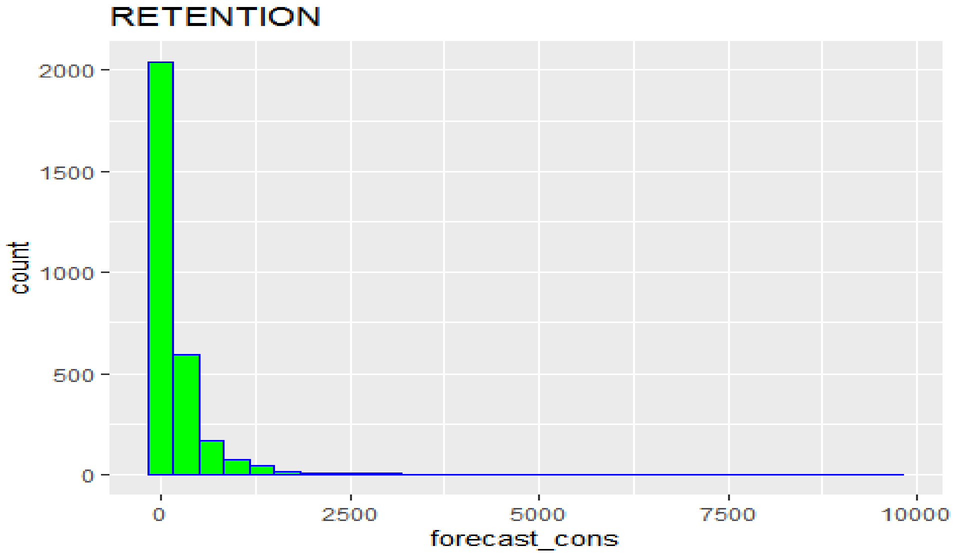
Total (retention + churn)

```
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).
```



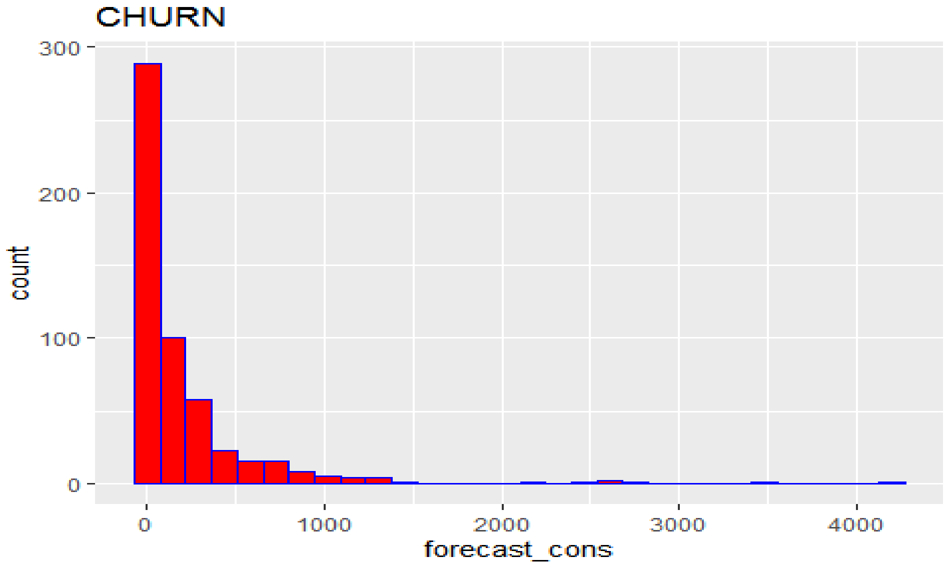
Histogram for forecast_cons (RETENTION)

```
forecast %>% filter(churn==0) %>%
   ggplot(aes(forecast_cons)) +
   geom_histogram(fill="green",color = I("blue")) +
   ggtitle("RETENTION")
```



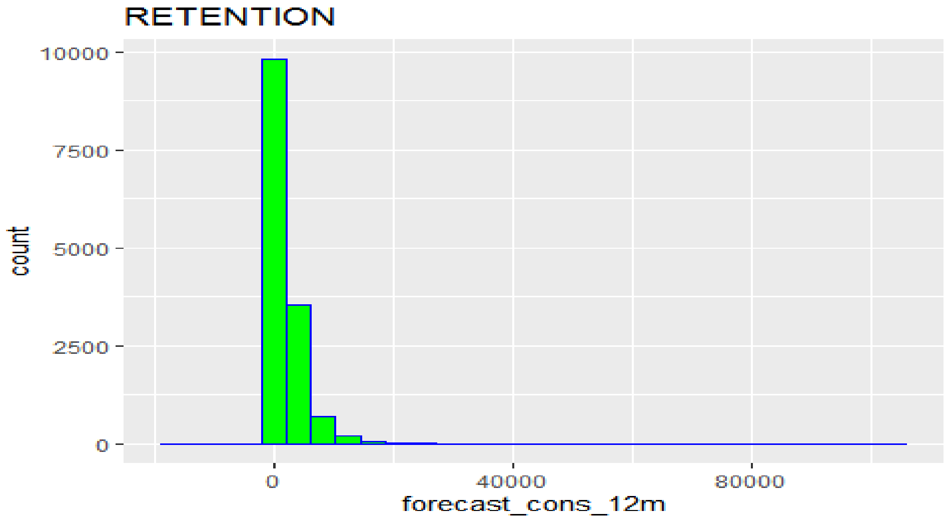
Histogram for forecast_cons (CHURN)

```
forecast %>% filter(churn==1) %>%
    ggplot(aes(forecast_cons)) +
    geom_histogram(fill="red",color = I("blue")) +
    ggtitle("CHURN")
```



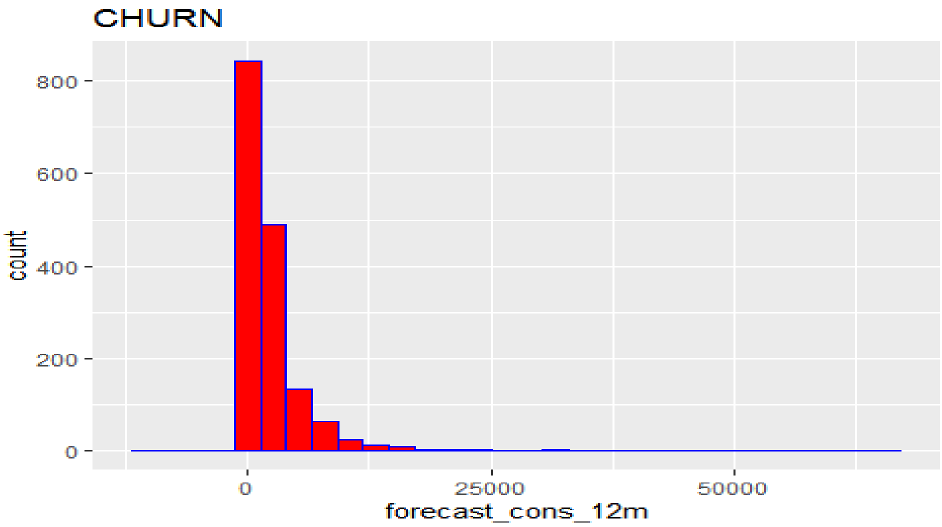
Histogram for forecast_cons_12m (RETENTION)

```
    forecast %>%
    filter(churn==0) %>%
    ggplot(aes(forecast_cons_12m)) +
    geom_histogram(fill="green",color = I("blue")) +
    ggtitle("RETENTION")
```



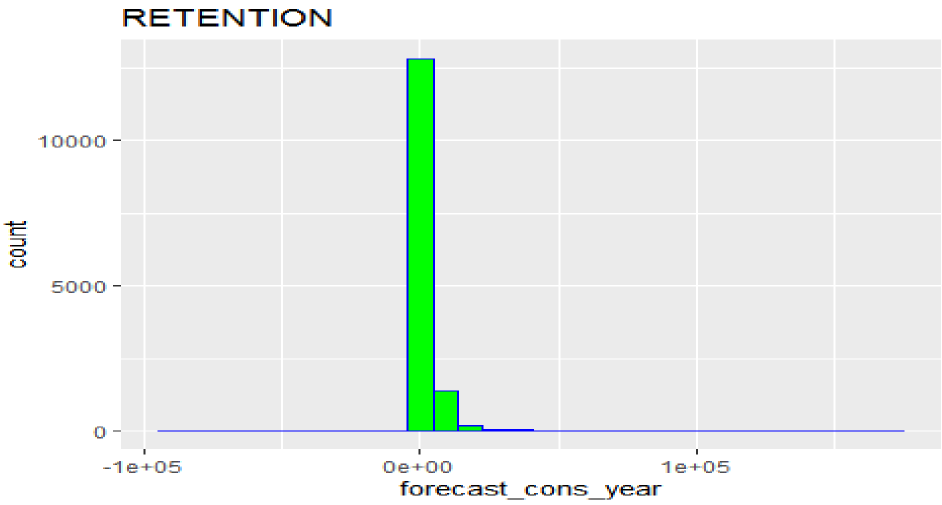
Histogram for forecast_cons_12m (CHURN)

```
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`.
```



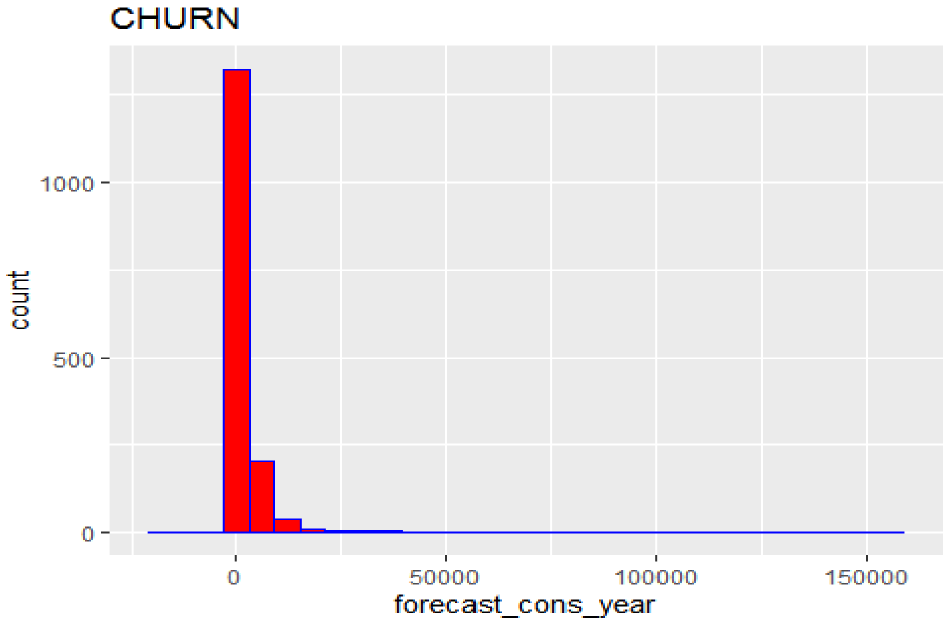
Histogram for forecast_cons_year (RETENTION)

```
   forecast %>%
   filter(churn==0) %>%
   ggplot(aes(forecast_cons_year)) +
   geom_histogram(fill="green",color = I("blue")) +
   ggtitle("RETENTION")
```



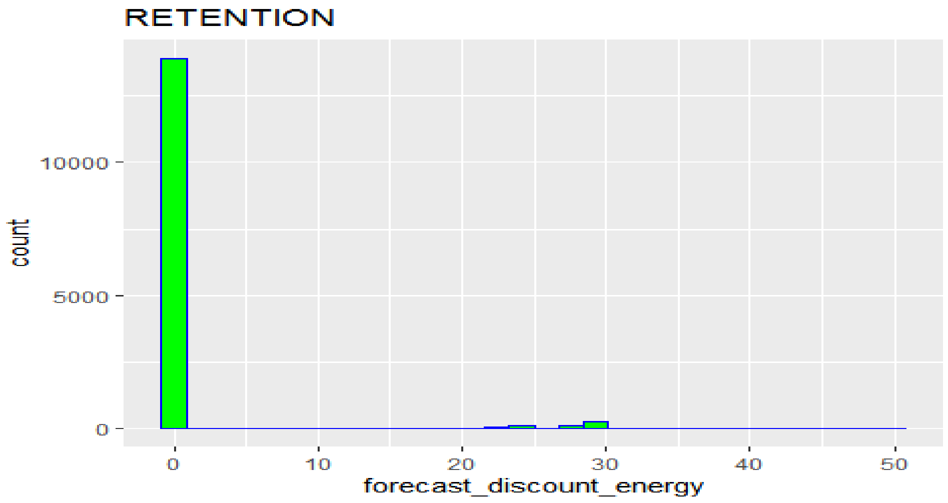
Histogram for forecast_cons_year (CHURN)

```
forecast %>%
filter(churn==1) %>%
ggplot(aes(forecast_cons_year)) +
geom_histogram(fill="red",color = I("blue")) +
ggtitle("CHURN")
```



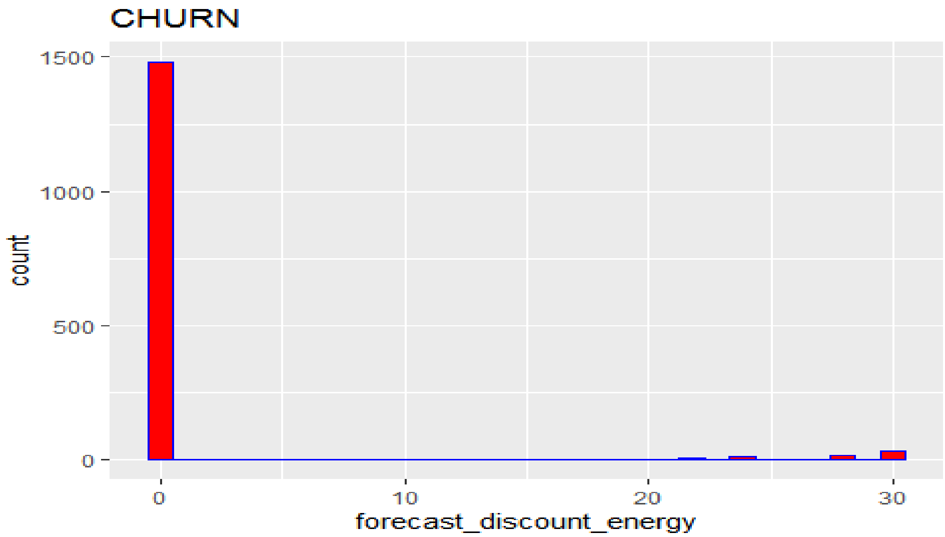
Histogram for forecast_discount_energy (RETENTION)

```
forecast %>%
filter(churn==0) %>%
ggplot(aes(forecast_discount_energy)) +
geom_histogram(fill="green",color = I("blue")) +
ggtitle("RETENTION")
```



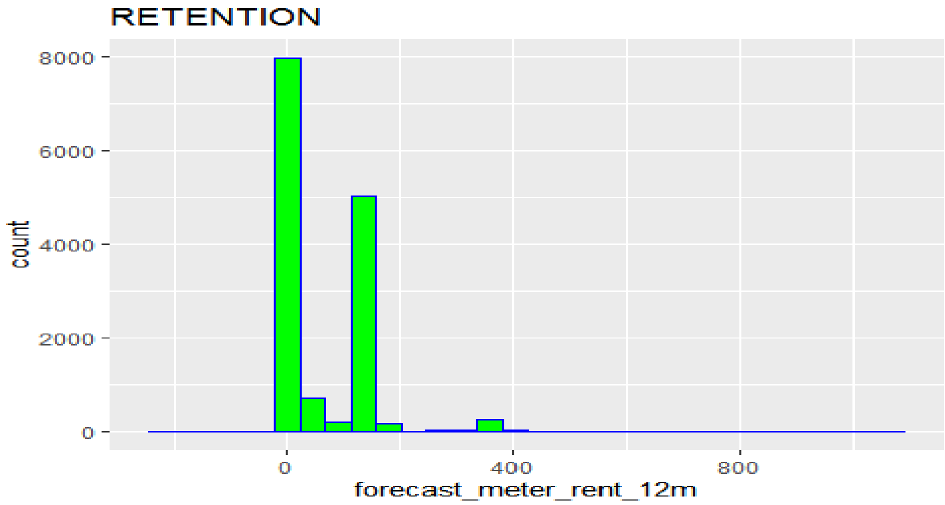
Histogram for forecast_discount_energy (CHURN)

```
   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)

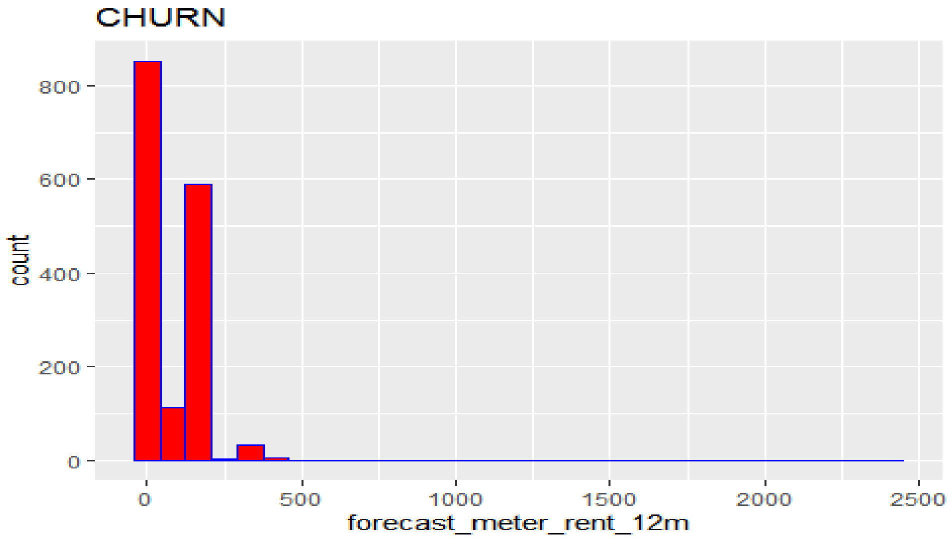
```
    forecast %>%
    filter(churn==0) %>%
    ggplot(aes(forecast_meter_rent_12m)) +
    geom_histogram(fill="green",color = I("blue")) +
    ggtitle("RETENTION")
```



Histogram for forecast_meter_rent_12m (CHURN)

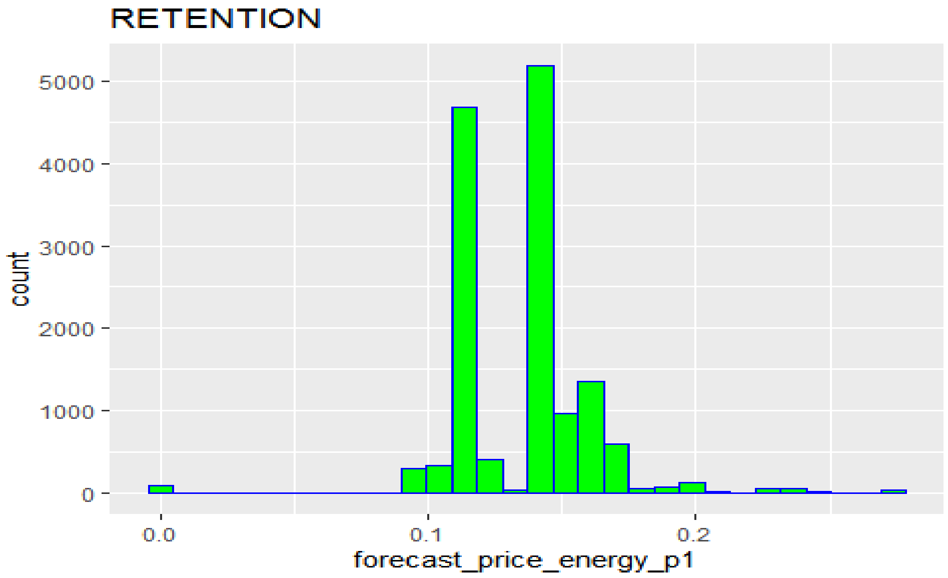
```
 forecast %>% filter(churn==1) %>%
  ggplot(aes(forecast_meter_rent_12m)) +
  geom_histogram(fill="red",color = I("blue")) +
  ggtitle("CHURN")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



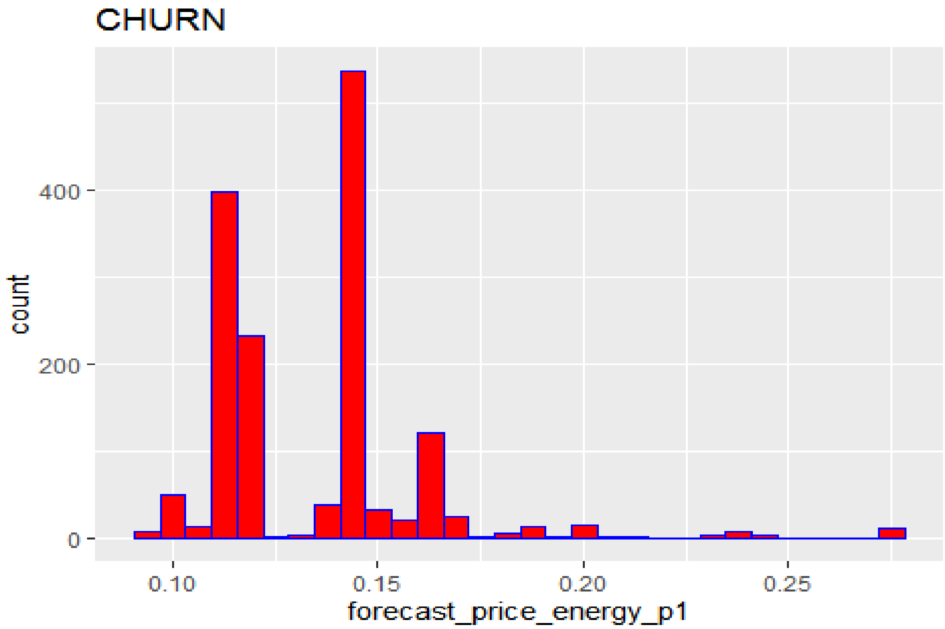
Histogram for forecast_price_energy_p1 (RETENTION)

```
  forecast %>%
  filter(churn==0) %>%
  ggplot(aes(forecast_price_energy_p1)) +
  geom_histogram(fill="green",color = I("blue")) +
  ggtitle("RETENTION")
```



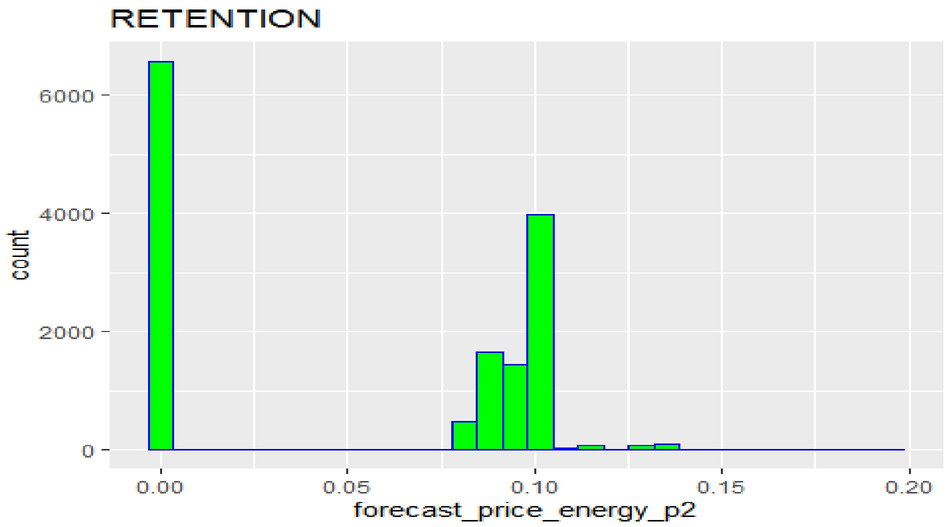
Histogram for forecast_price_energy_p1 (CHURN)

```
forecast %>% filter(churn==1) %>%
  ggplot(aes(forecast_price_energy_p1)) +
  geom_histogram(fill="red",color = I("blue")) +
  ggtitle("CHURN")
```



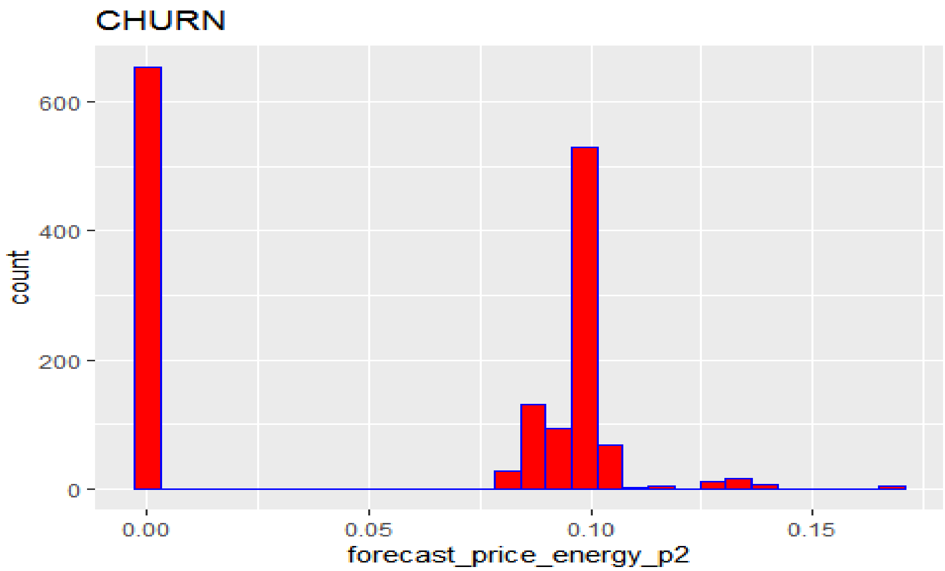
Histogram for forecast_price_energy_p2 (RETENTION)

```
  forecast %>%
  filter(churn==0) %>%
  ggplot(aes(forecast_price_energy_p2)) +
  geom_histogram(fill="green",color = I("blue")) +
  ggtitle("RETENTION")
```



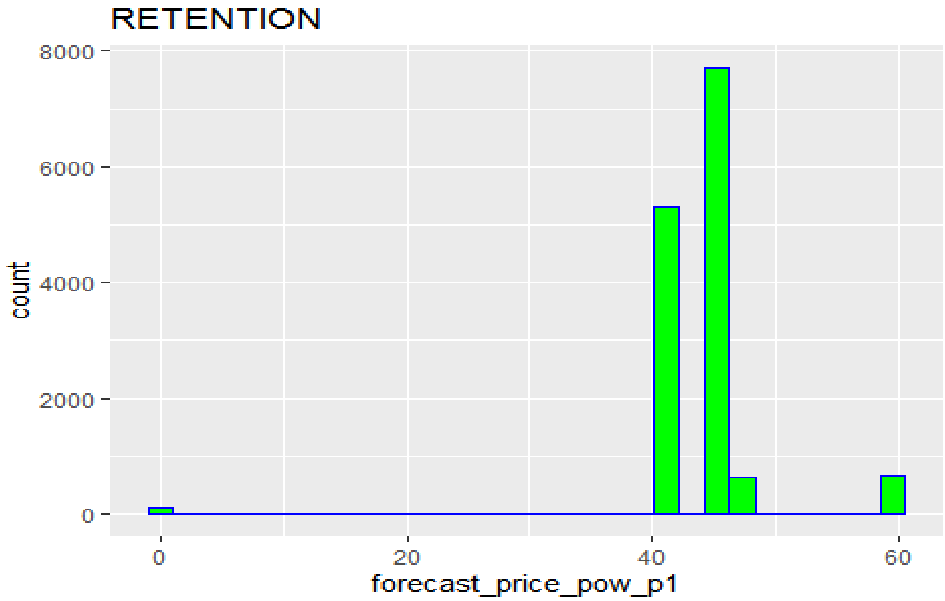
Histogram for forecast_price_energy_p2 (CHURN)

```
forecast %>% filter(churn==1) %>%
  ggplot(aes(forecast_price_energy_p2)) +
  geom_histogram(fill="red",color = I("blue")) +
  ggtitle("CHURN")
```



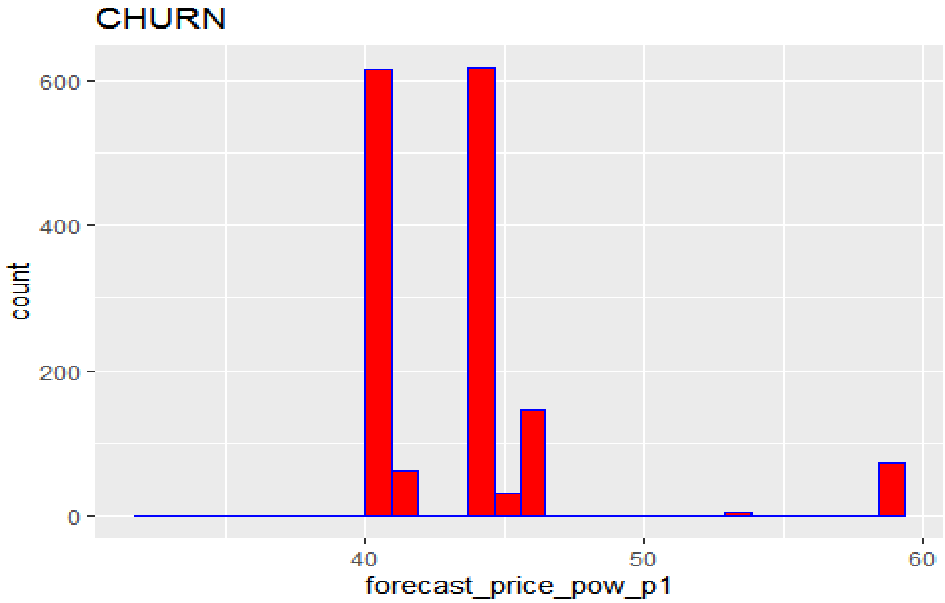
Histogram for forecast_price_pow_p1 (RETENTION)

```
  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

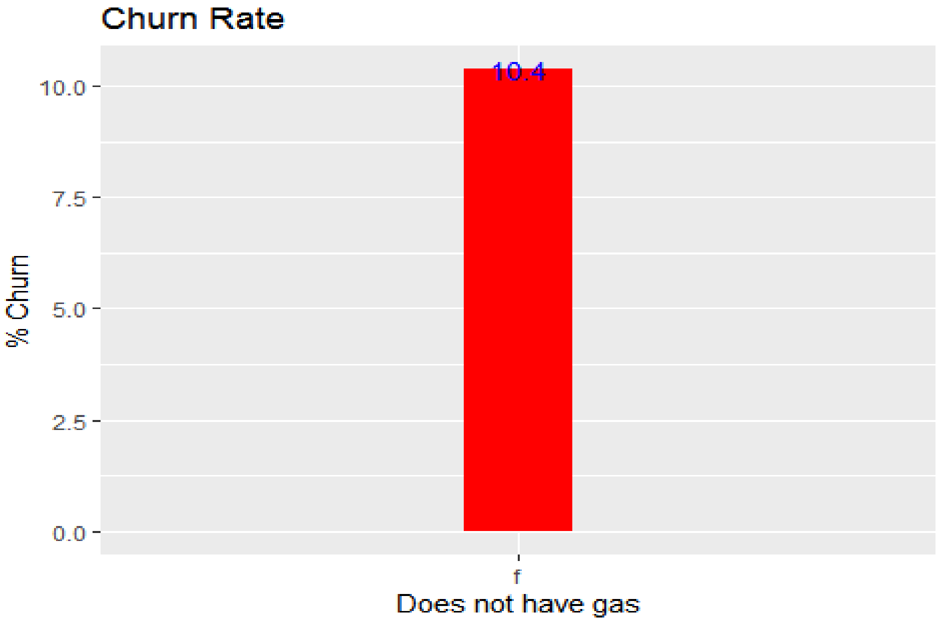
```
  contract_type = train %>%
                  select(id, has_gas, churn)
                  class(contract_type)

## [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)
```

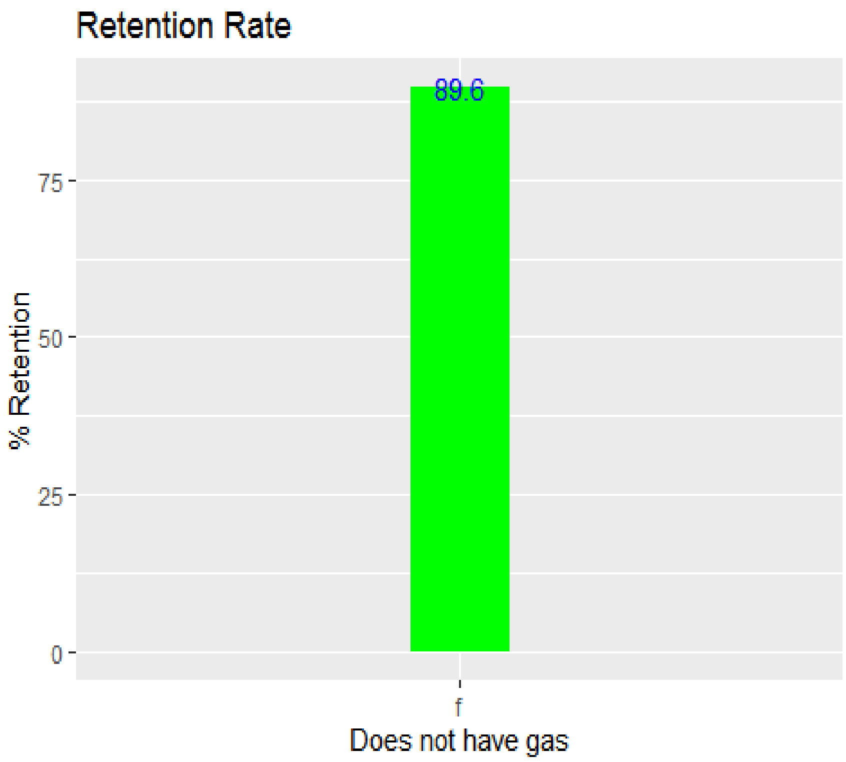
Visualization for has_gas= "f" which Churned

```
  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)
```

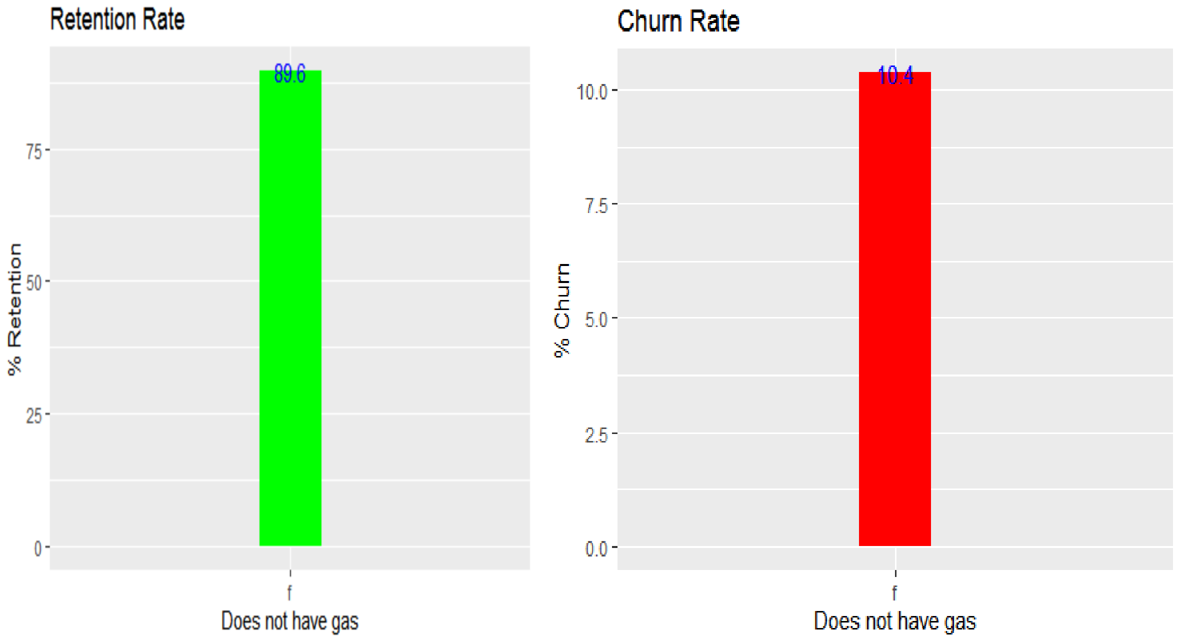


Visualization for has_gas= "f" which Retained

```
    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)
```

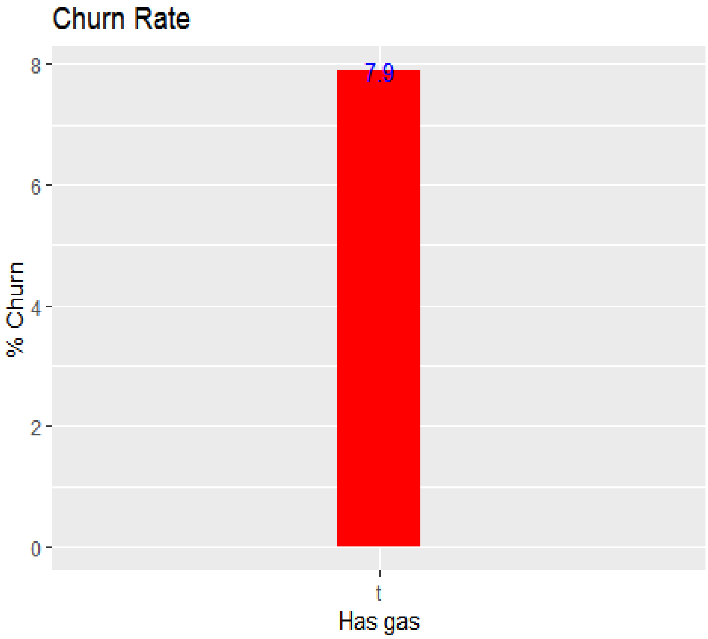


COMPARISION BETWEEN RETENTION AND CHURN RATES WHICH DOES NOT "HAVE_GAS"



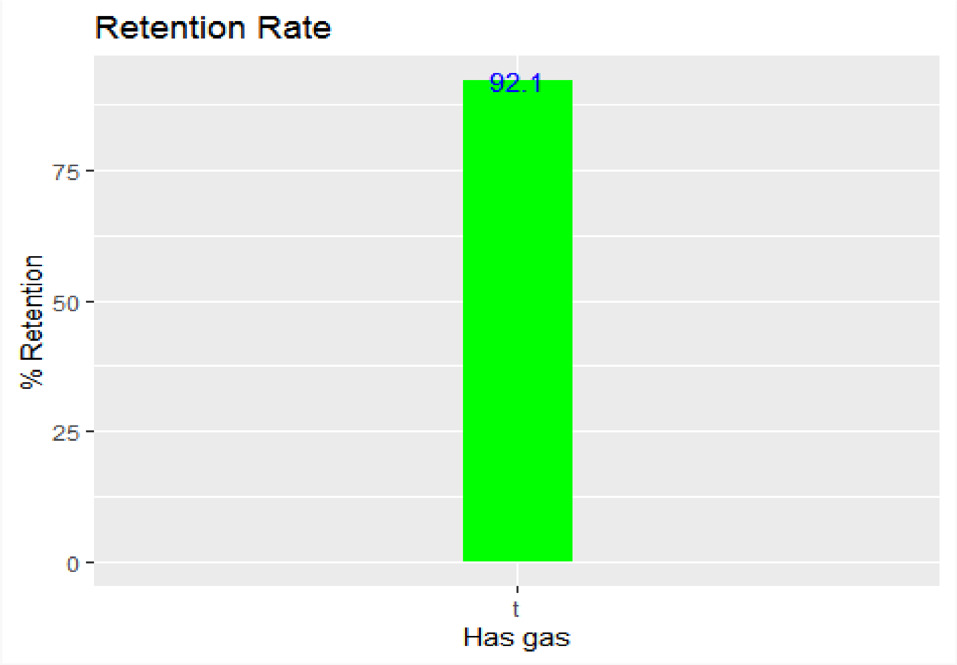
Visualization for has_gas= "t" which Churned

```
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)
```

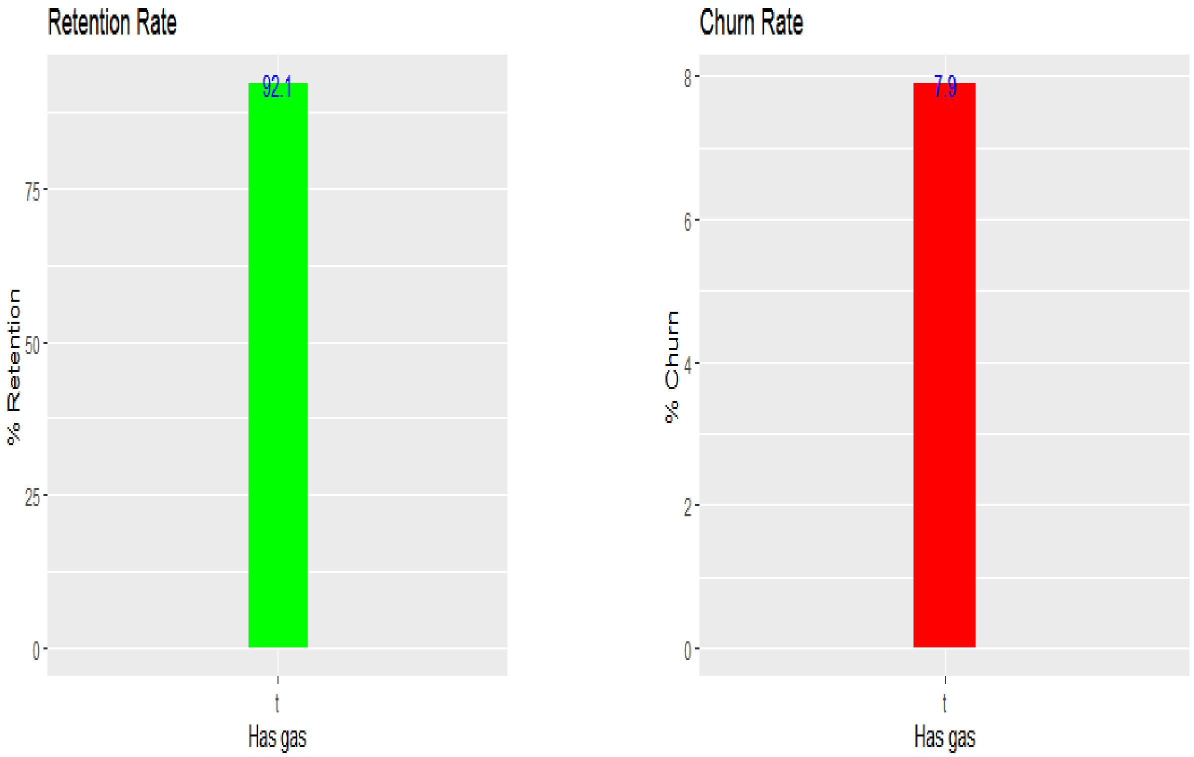


Visualization for has_gas = "t" which Retained

```
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”

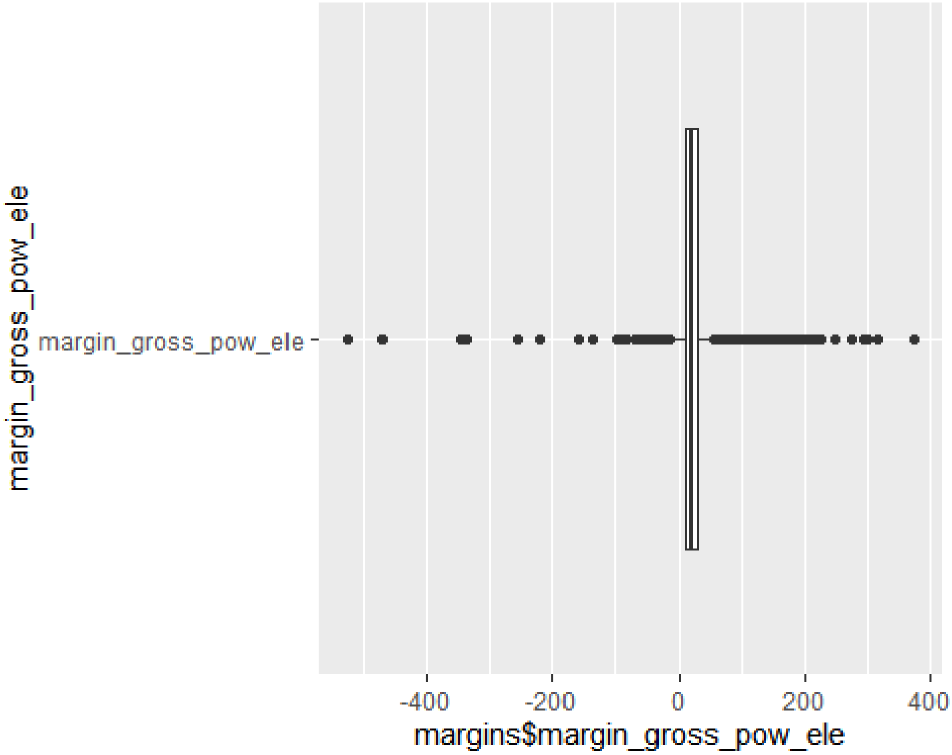


MARGINS

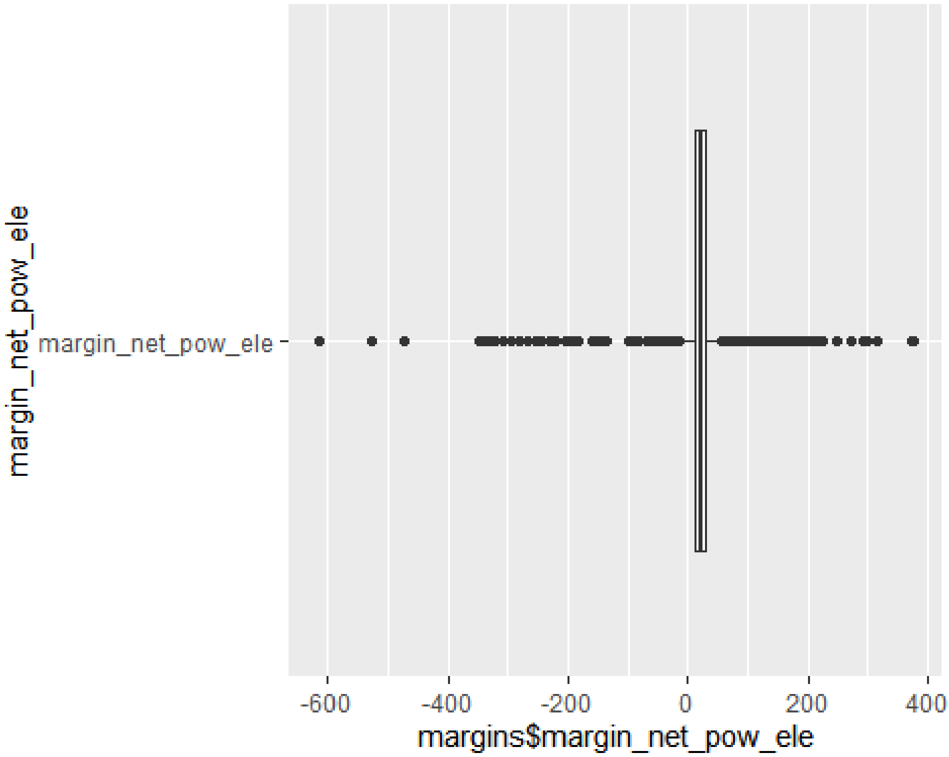
```
  margins = train %>%
            select(id, margin_gross_pow_ele,margin_net_pow_ele,net_margin)
  colSums(is.na(margins))

##                   id margin_gross_pow_ele   margin_net_pow_ele
##                    0                   13                   13
##           net_margin
##                   15
```

Box Plots

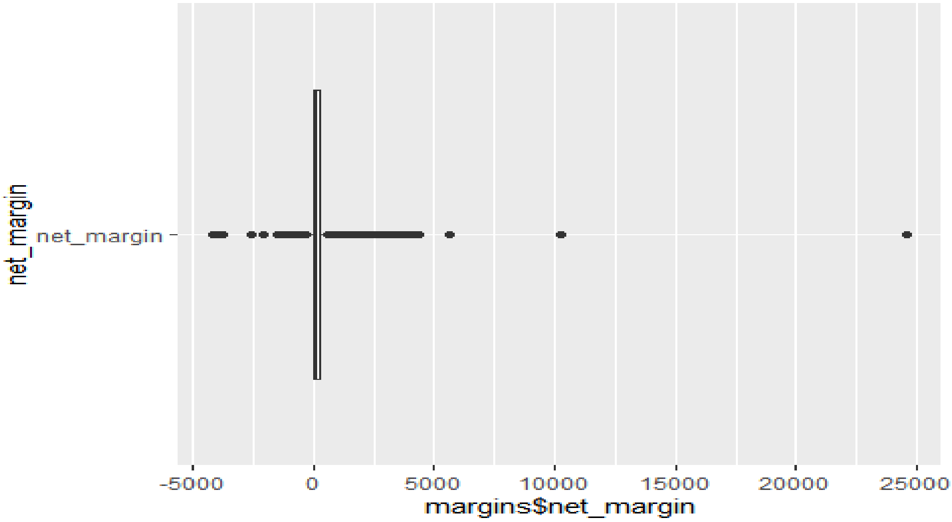
```
  qplot("margin_gross_pow_ele", margins$margin_gross_pow_ele, geom =
"boxplot") + coord_flip()

## 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).
```

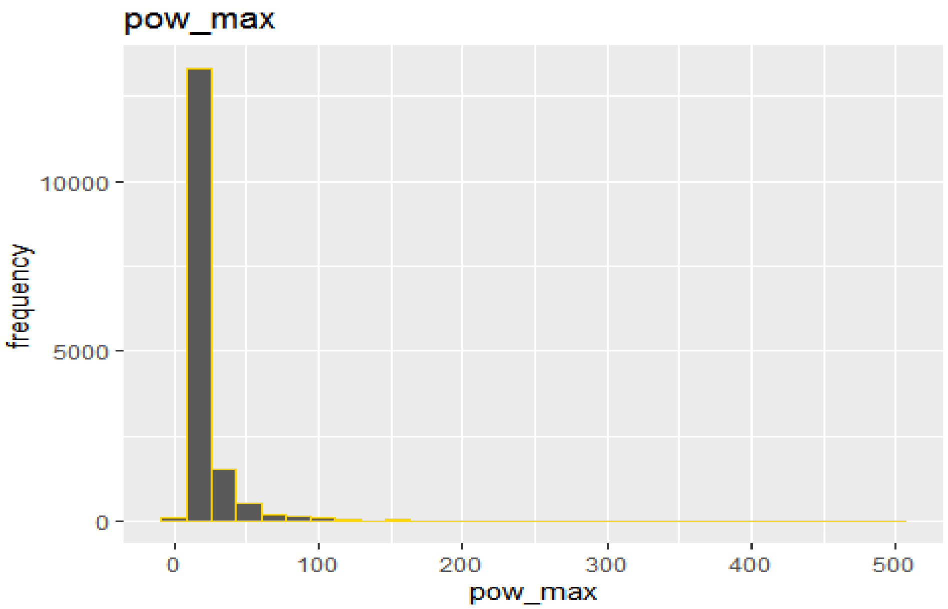


SUBSCRIBED POWER

```
power = train %>%
        select(id, pow_max, churn)

colSums(is.na(power))

##       id pow_max   churn
##        0       3       0
```

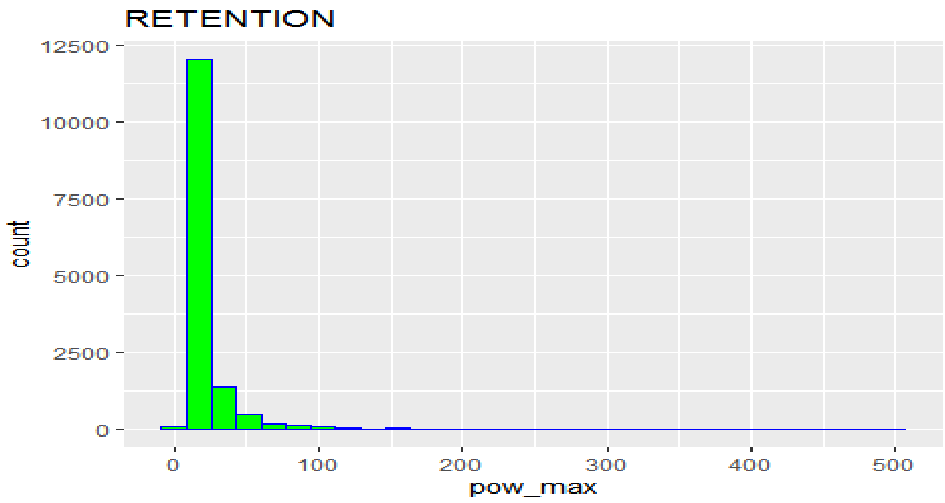
Total (retention + churn)

```
  qplot(power$pow_max, geom = "histogram",
        color = I("GOLD"),
        xlab = "pow_max",
        ylab = "frequency",
        main = "pow_max")
```



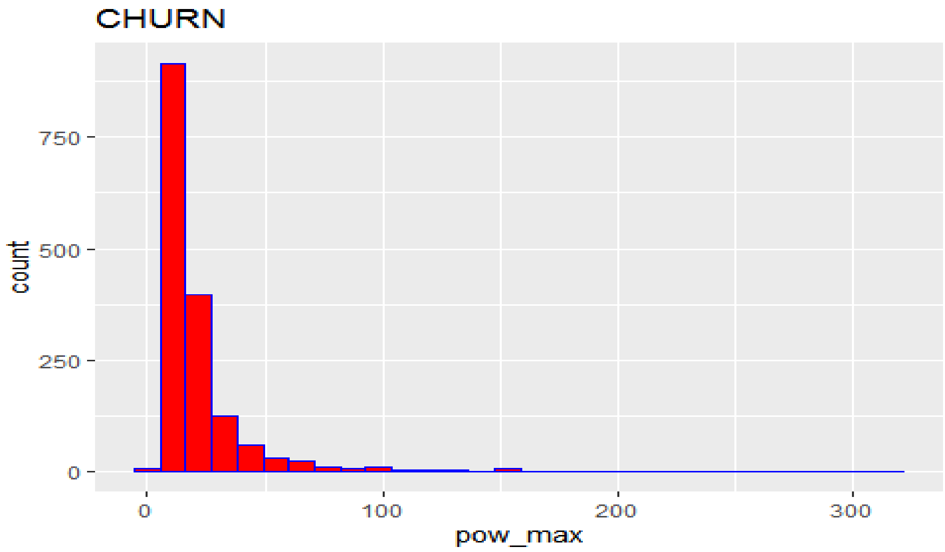
Histogram for pow_max (RETENTION)

```
  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)
```

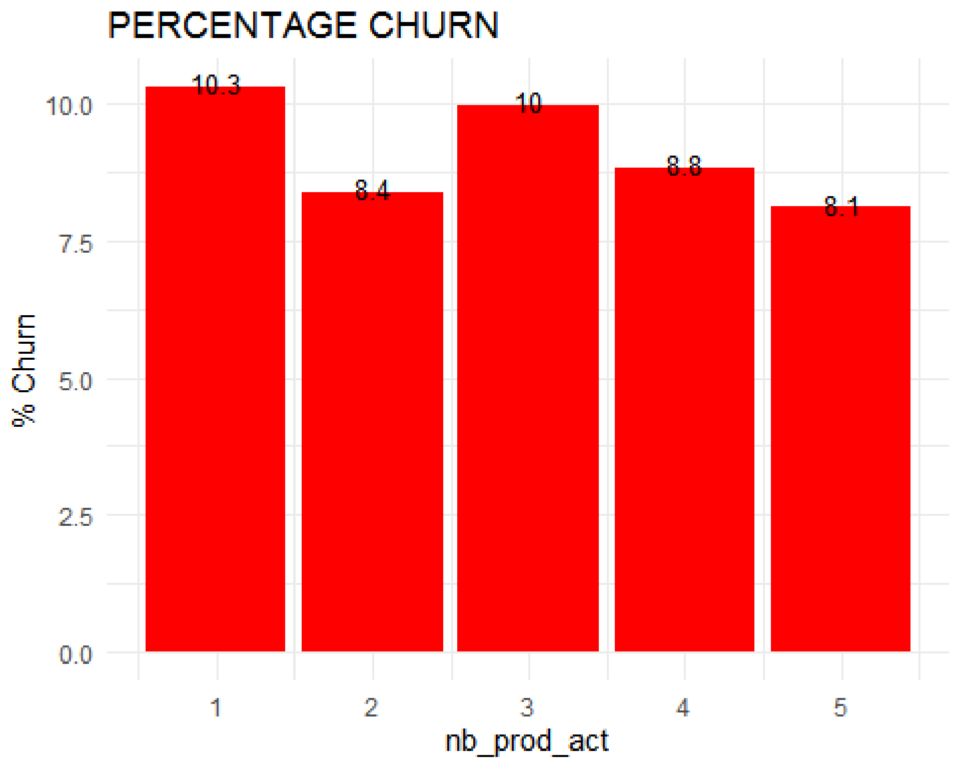
```
  others_1[is.na(others_1)]=0  ## to replace NA's with Zero's

  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)
```

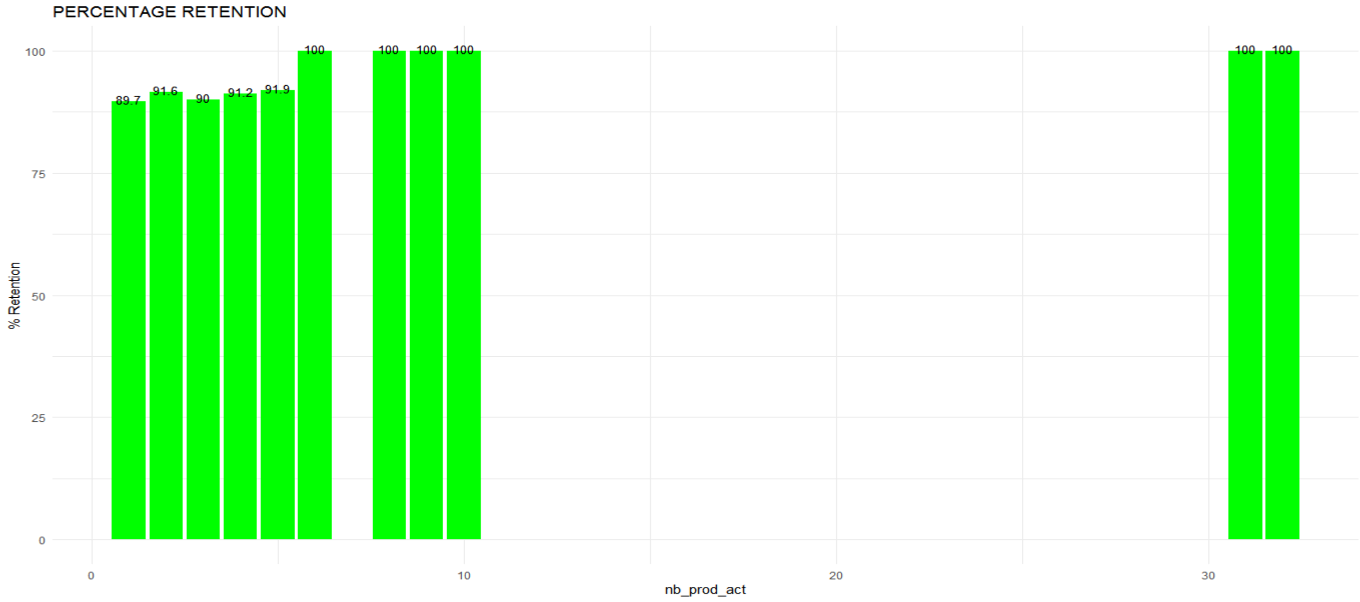
Visualization for Churn

```
    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()
```

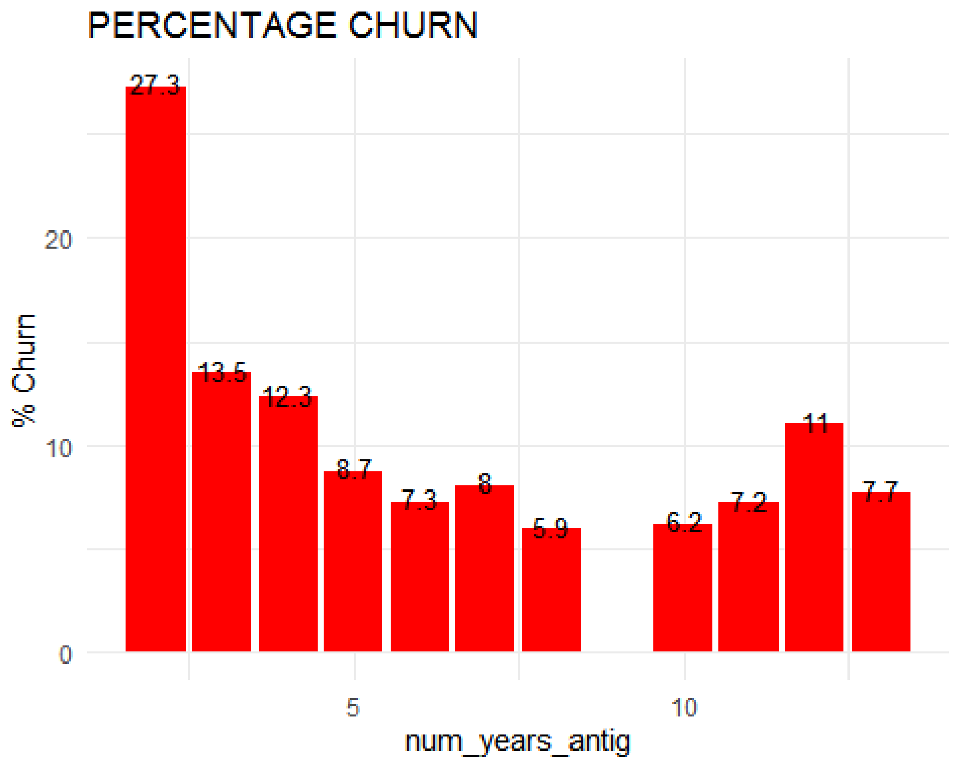


num_years_antig

```
  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)
```

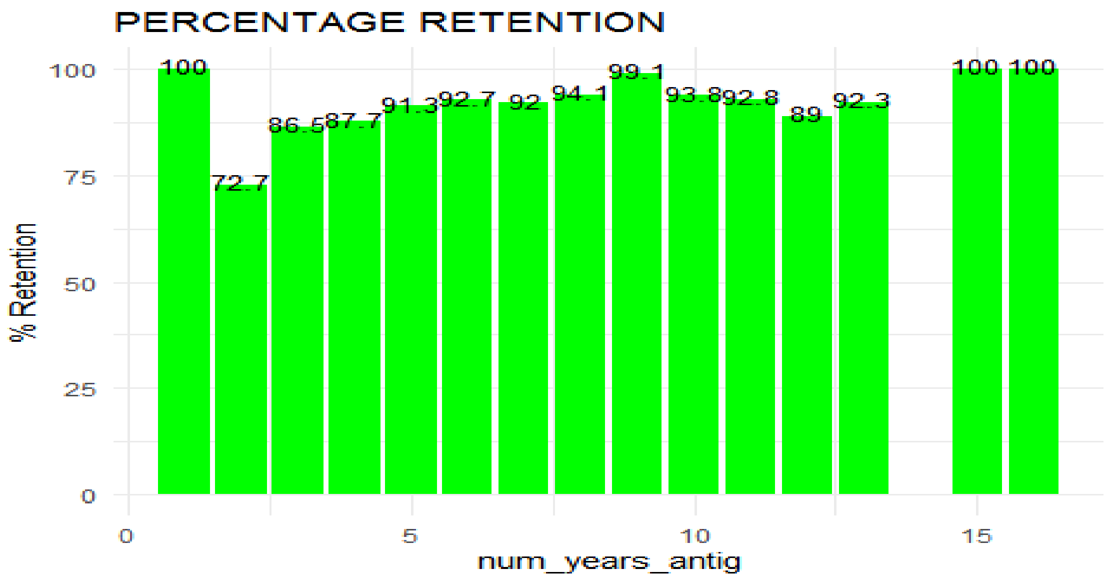
Visualization for Churn

```
  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()
```

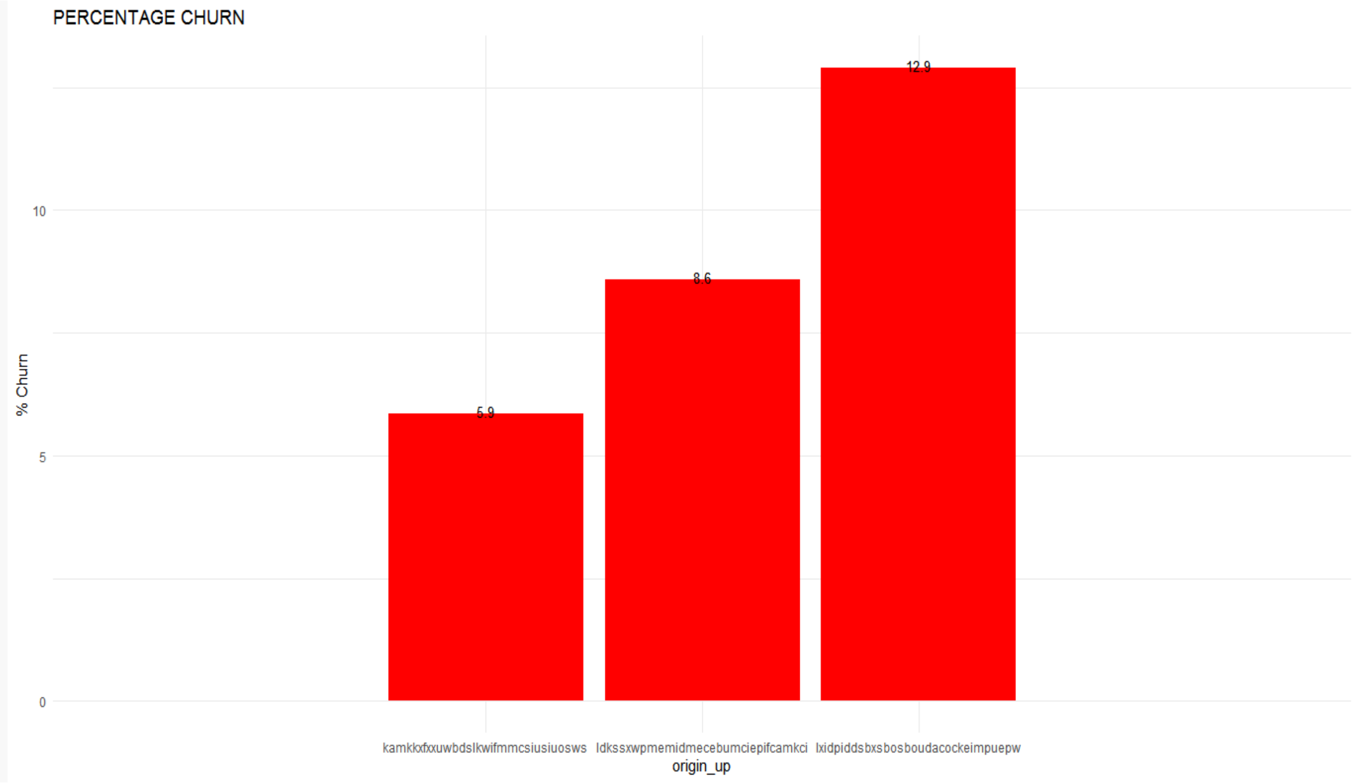


origin_up

```
  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)
```

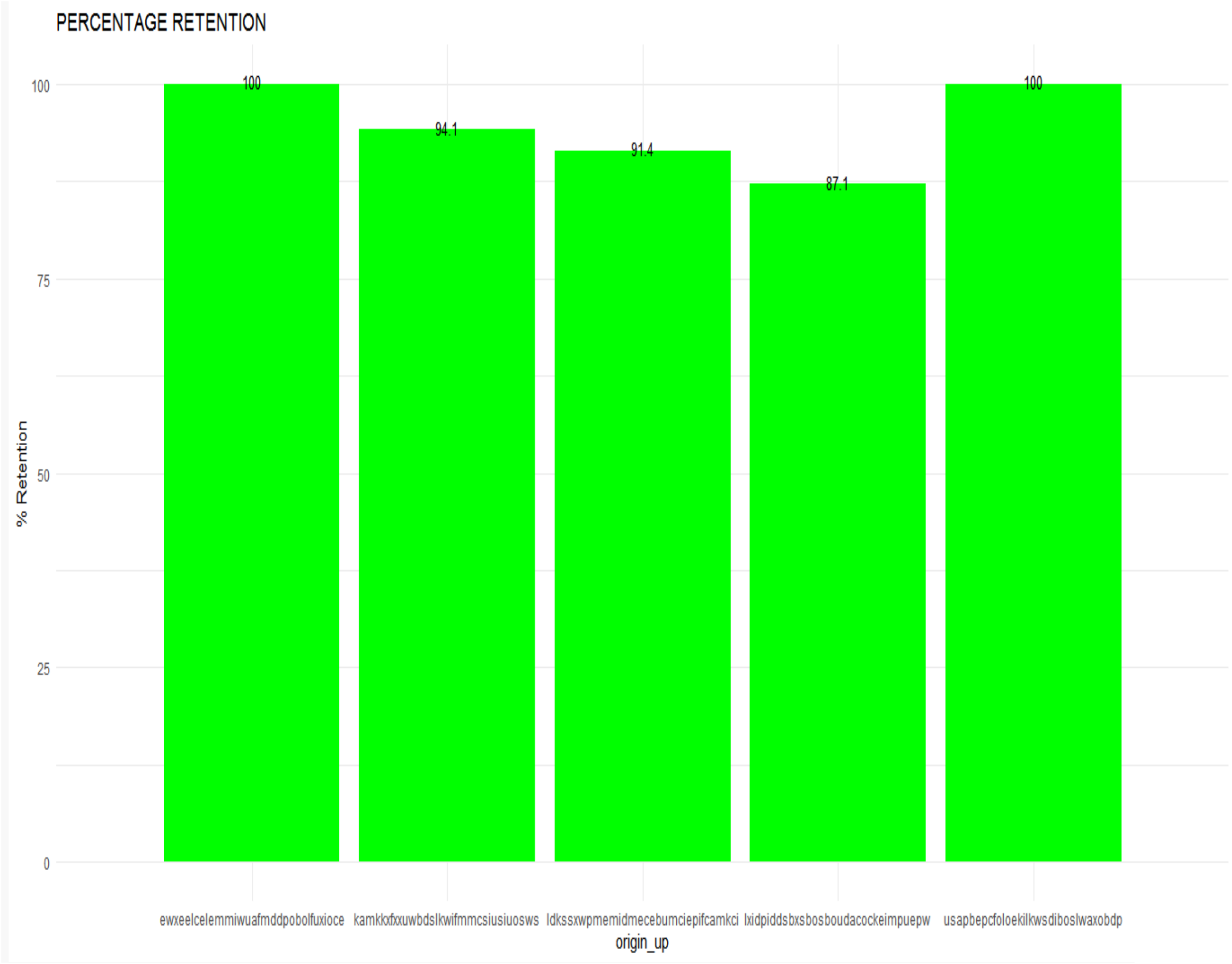
Bar Plot Visualization for Churn

```
    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
```

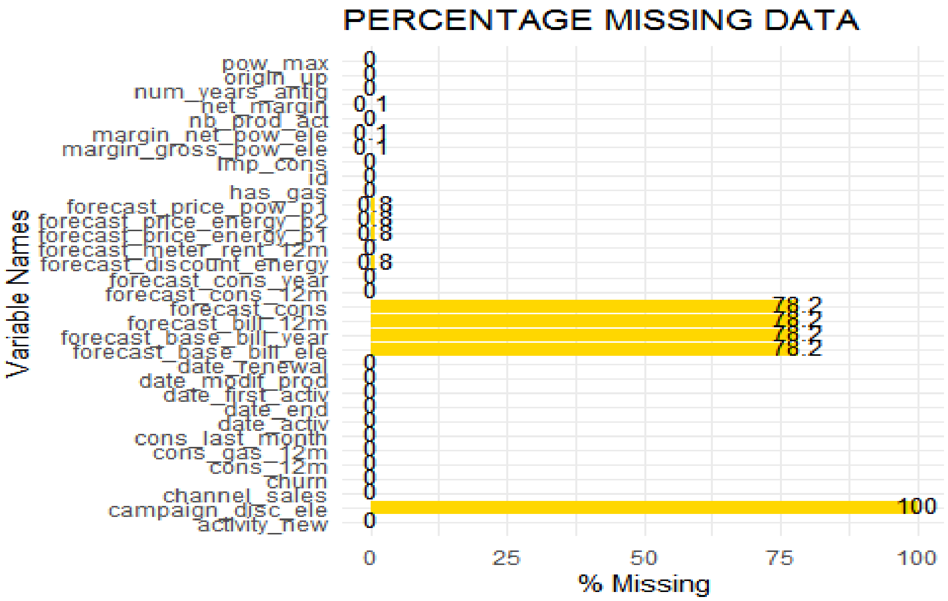
Changing dates into date format

```
 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

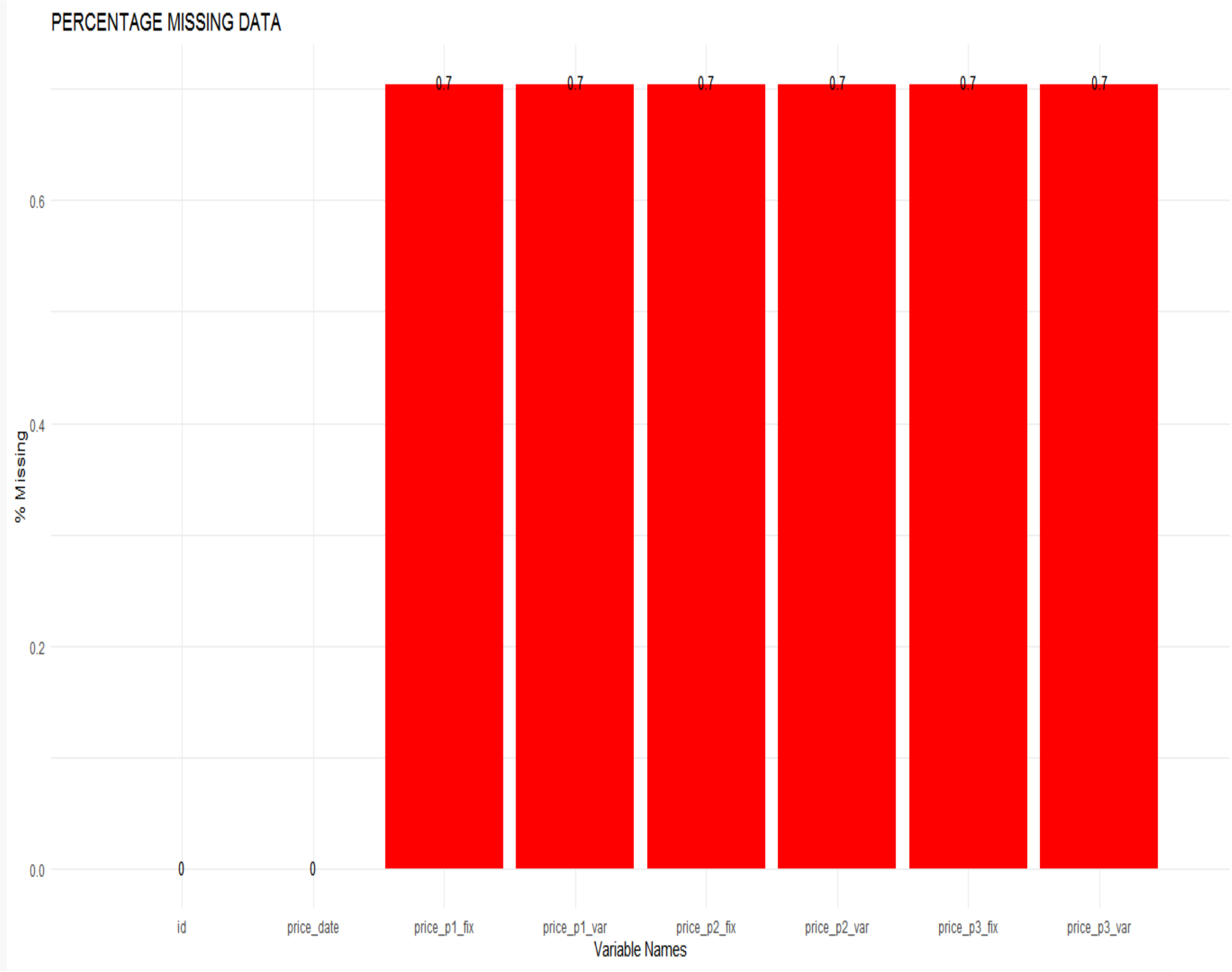
```
   train_3 = train_2

  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

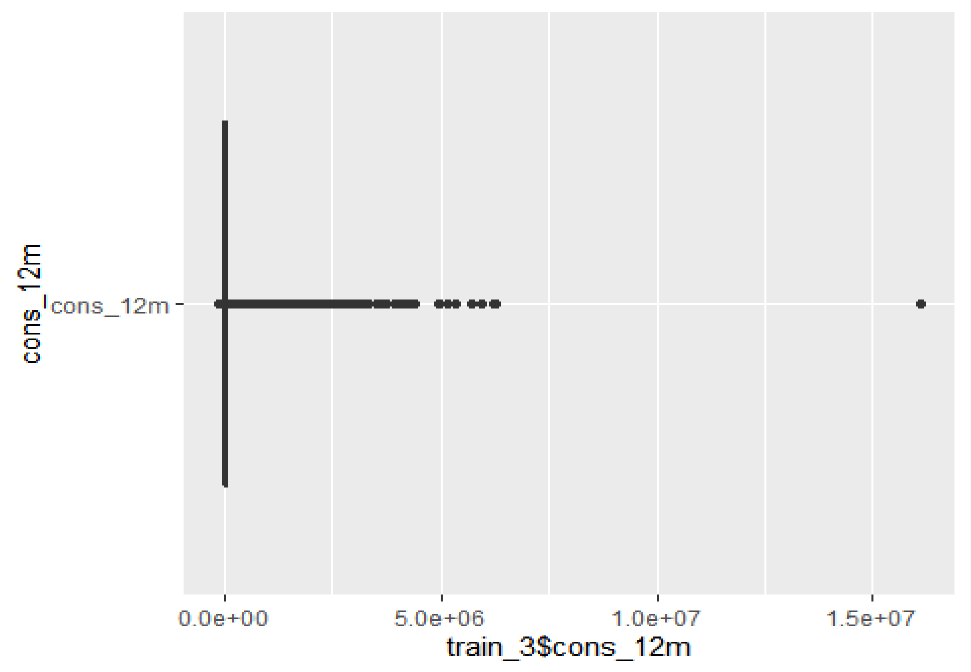
```
 var1Case = median(na.omit(p_1)$price_p1_var)
 var2Case = median(na.omit(p_1)$price_p2_var)
 var3Case = median(na.omit(p_1)$price_p3_var)

 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

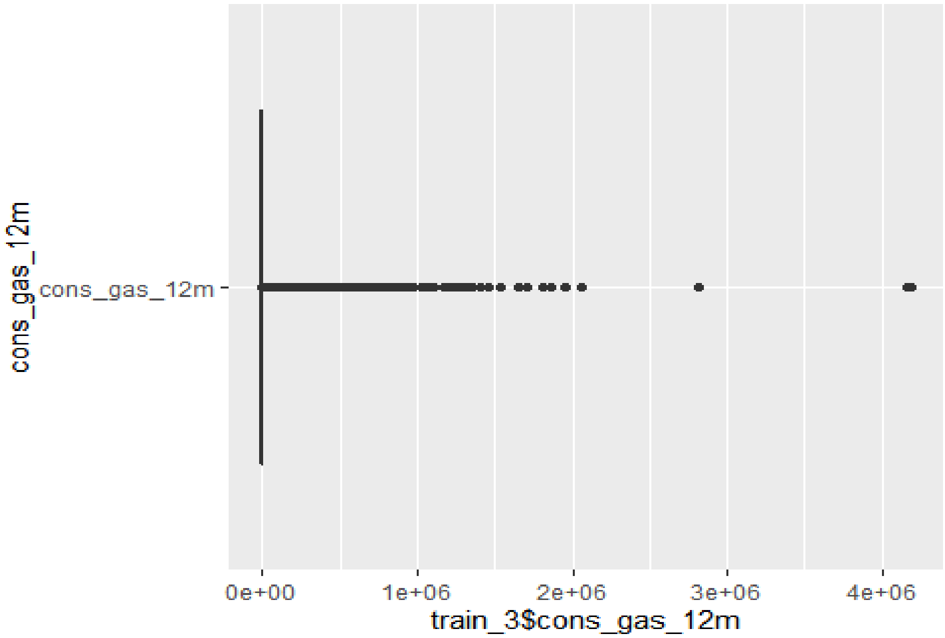
```
p_1 = p_1 %>%
   mutate(price_p1_var_complete = case_when(is.na(price_p1_var) ~ 
var1Case,TRUE ~ price_p1_var),
          price_p2_var_complete = case_when(is.na(price_p2_var) ~ 
var2Case,TRUE ~ price_p2_var),
          price_p3_var_complete = case_when(is.na(price_p3_var) ~ 
var3Case,TRUE ~ price_p3_var),
          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

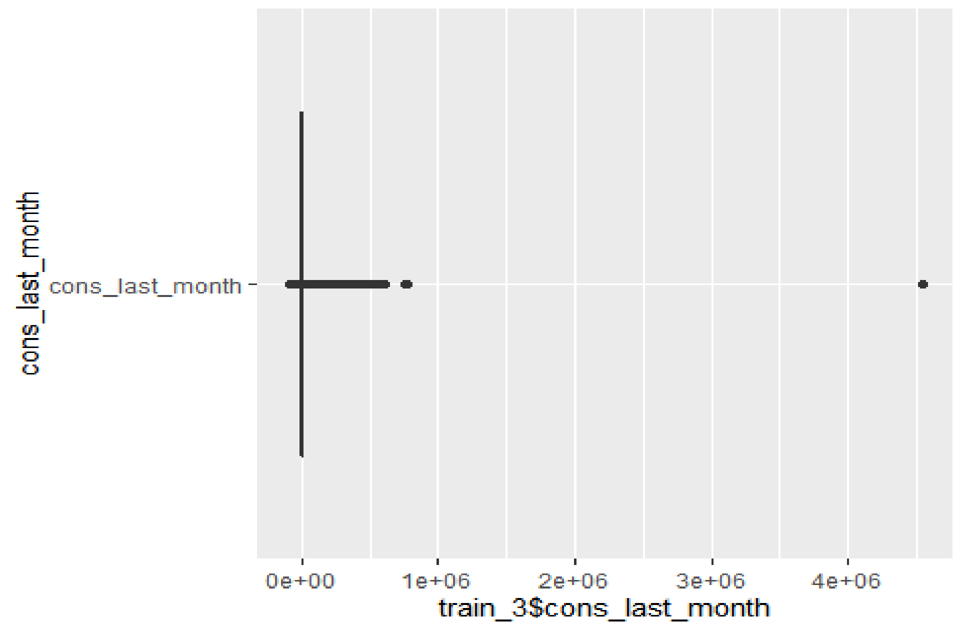
```
  qplot("cons_12m", train_3$cons_12m, geom = "boxplot") + coord_flip()
```



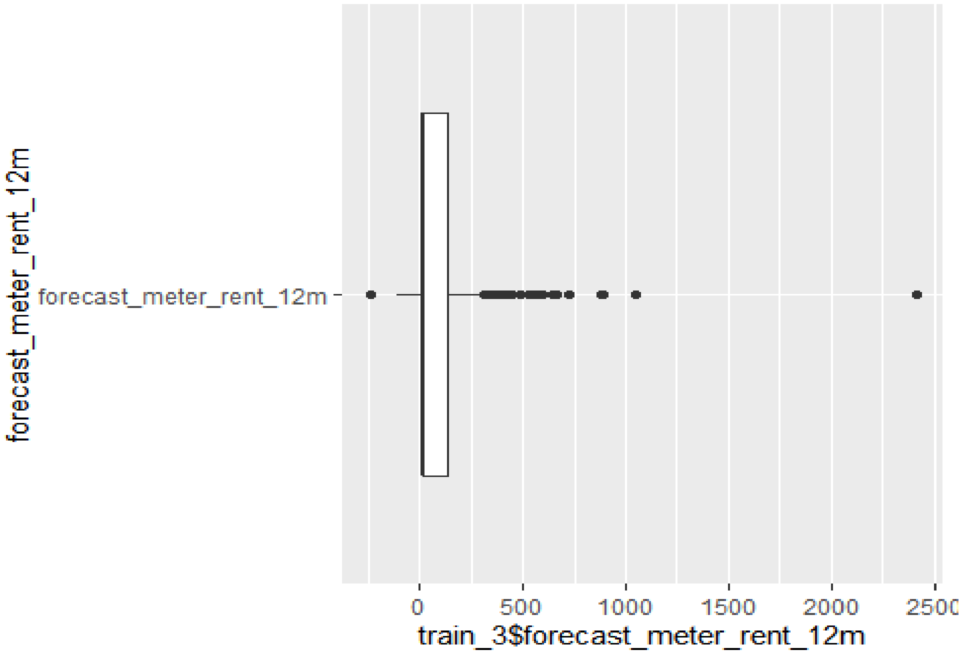
```
  qplot("cons_gas_12m", train_3$cons_gas_12m, geom = "boxplot") +
coord_flip()
```



```
  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)
```