Project

Varun Jain

14 Feb 2019

1.Introduction

For an Auto Insurance company, predict the customer lifetime value (CLV). CLV is the total revenue the client will derive from their entire relationship with a customer. Because we don't know how long each customer relationship will be, we make a good estimate and state CLV as a periodic value

```
#Import data from Excel
library(readxl)
setwd("D:/Project/Insurance value")
df = read_excel("Insurance_Marketing-Customer-Value-Analysis.xlsx")
head(df)
```

```
## # A tibble: 6 x 24
##   Customer State `Customer Lifet~ Response Coverage Education
##   <chr>    <chr>            <dbl> <chr>    <chr>    <chr>
## 1 BU79786  Wash~            2764. No       Basic    Bachelor
## 2 QZ44356  Ariz~            6980. No       Extended Bachelor
## 3 AI49188  Neva~           12887. No       Premium  Bachelor
## 4 WW63253  Cali~            7646. No       Basic    Bachelor
## 5 HB64268  Wash~            2814. No       Basic    Bachelor
## 6 OC83172  Oreg~            8256. Yes      Basic    Bachelor
## # ... with 18 more variables: `Effective To Date` <dttm>,
## #   EmploymentStatus <chr>, Gender <chr>, Income <dbl>, `Location
## #   Code` <chr>, `Marital Status` <chr>, `Monthly Premium Auto` <dbl>,
## #   `Months Since Last Claim` <dbl>, `Months Since Policy
## #   Inception` <dbl>, `Number of Open Complaints` <dbl>, `Number of
## #   Policies` <dbl>, `Policy Type` <chr>, Policy <chr>, `Renew Offer
## #   Type` <chr>, `Sales Channel` <chr>, `Total Claim Amount` <dbl>,
## #   `Vehicle Class` <chr>, `Vehicle Size` <chr>
```

2. Visualisation Data

```
dim(df)
```

```
## [1] 9134   24
```

```
#Data contain 9134 rows and 24 columns
```

```r
# Check missing values
colSums(is.na(df))
```

```
##                      Customer                         State
##                             0                             0
##       Customer Lifetime Value                      Response
##                             0                             0
##                      Coverage                     Education
##                             0                             0
##             Effective To Date              EmploymentStatus
##                             0                             0
##                        Gender                        Income
##                             0                             0
##                 Location Code                Marital Status
##                             0                             0
##          Monthly Premium Auto       Months Since Last Claim
##                             0                             0
## Months Since Policy Inception     Number of Open Complaints
##                             0                             0
##            Number of Policies                   Policy Type
##                             0                             0
##                        Policy              Renew Offer Type
##                             0                             0
##                 Sales Channel            Total Claim Amount
##                             0                             0
##                 Vehicle Class                  Vehicle Size
##                             0                             0
```

```r
str(df)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    9134 obs. of  24 variables:
##  $ Customer                     : chr  "BU79786" "QZ44356" "AI49188"
"WW63253" ...
##  $ State                        : chr  "Washington" "Arizona" "Nevada"
"California" ...
##  $ Customer Lifetime Value      : num  2764 6980 12887 7646 2814 ...
##  $ Response                     : chr  "No" "No" "No" "No" ...
##  $ Coverage                     : chr  "Basic" "Extended" "Premium"
"Basic" ...
##  $ Education                    : chr  "Bachelor" "Bachelor" "Bachelor"
"Bachelor" ...
##  $ Effective To Date            : POSIXct, format: "2011-02-24" "2011-01-
31" ...
##  $ EmploymentStatus             : chr  "Employed" "Unemployed" "Employed"
"Unemployed" ...
##  $ Gender                       : chr  "F" "F" "F" "M" ...
##  $ Income                       : num  56274 0 48767 0 43836 ...
##  $ Location Code                : chr  "Suburban" "Suburban" "Suburban"
"Suburban" ...
##  $ Marital Status               : chr  "Married" "Single" "Married"
```

```
"Married" ...
##  $ Monthly Premium Auto         : num  69 94 108 106 73 69 67 101 71 93
...
##  $ Months Since Last Claim      : num  32 13 18 18 12 14 0 0 13 17 ...
##  $ Months Since Policy Inception: num  5 42 38 65 44 94 13 68 3 7 ...
##  $ Number of Open Complaints    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Number of Policies           : num  1 8 2 7 1 2 9 4 2 8 ...
##  $ Policy Type                  : chr  "Corporate Auto" "Personal Auto"
"Personal Auto" "Corporate Auto" ...
##  $ Policy                       : chr  "Corporate L3" "Personal L3"
"Personal L3" "Corporate L2" ...
##  $ Renew Offer Type             : chr  "Offer1" "Offer3" "Offer1" "Offer1"
...
##  $ Sales Channel                : chr  "Agent" "Agent" "Agent" "Call
Center" ...
##  $ Total Claim Amount           : num  385 1131 566 530 138 ...
##  $ Vehicle Class                : chr  "Two-Door Car" "Four-Door Car"
"Two-Door Car" "SUV" ...
##  $ Vehicle Size                 : chr  "Medsize" "Medsize" "Medsize"
"Medsize" ...
```

```
summary(df)
```

```
##    Customer            State           Customer Lifetime Value
##  Length:9134        Length:9134        Min.   : 1898
##  Class :character   Class :character   1st Qu.: 3994
##  Mode  :character   Mode  :character   Median : 5780
##                                        Mean   : 8005
##                                        3rd Qu.: 8962
##                                        Max.   :83325
##    Response           Coverage          Education
##  Length:9134        Length:9134        Length:9134
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Effective To Date             EmploymentStatus      Gender
##  Min.   :2011-01-01 00:00:00   Length:9134        Length:9134
##  1st Qu.:2011-01-15 00:00:00   Class :character   Class :character
##  Median :2011-01-29 00:00:00   Mode  :character   Mode  :character
##  Mean   :2011-01-29 20:06:21
##  3rd Qu.:2011-02-13 00:00:00
##  Max.   :2011-02-28 00:00:00
##      Income      Location Code      Marital Status
##  Min.   :    0   Length:9134        Length:9134
##  1st Qu.:    0   Class :character   Class :character
##  Median :33890   Mode  :character   Mode  :character
##  Mean   :37657
##  3rd Qu.:62320
```

```
##  Max.   :99981
##  Monthly Premium Auto Months Since Last Claim
##  Min.   : 61.00       Min.   : 0.0
##  1st Qu.: 68.00       1st Qu.: 6.0
##  Median : 83.00       Median :14.0
##  Mean   : 93.22       Mean   :15.1
##  3rd Qu.:109.00       3rd Qu.:23.0
##  Max.   :298.00       Max.   :35.0
##  Months Since Policy Inception Number of Open Complaints
##  Min.   : 0.00                 Min.   :0.0000
##  1st Qu.:24.00                 1st Qu.:0.0000
##  Median :48.00                 Median :0.0000
##  Mean   :48.06                 Mean   :0.3844
##  3rd Qu.:71.00                 3rd Qu.:0.0000
##  Max.   :99.00                 Max.   :5.0000
##  Number of Policies Policy Type          Policy
##  Min.   :1.000      Length:9134        Length:9134
##  1st Qu.:1.000      Class :character   Class :character
##  Median :2.000      Mode  :character   Mode  :character
##  Mean   :2.966
##  3rd Qu.:4.000
##  Max.   :9.000
##  Renew Offer Type   Sales Channel      Total Claim Amount
##  Length:9134        Length:9134        Min.   :   0.099
##  Class :character   Class :character   1st Qu.: 272.258
##  Mode  :character   Mode  :character   Median : 383.945
##                                        Mean   : 434.089
##                                        3rd Qu.: 547.515
##                                        Max.   :2893.240
##  Vehicle Class      Vehicle Size
##  Length:9134        Length:9134
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

2.1 Data manipulation

```
#state
unique(df$State)
```

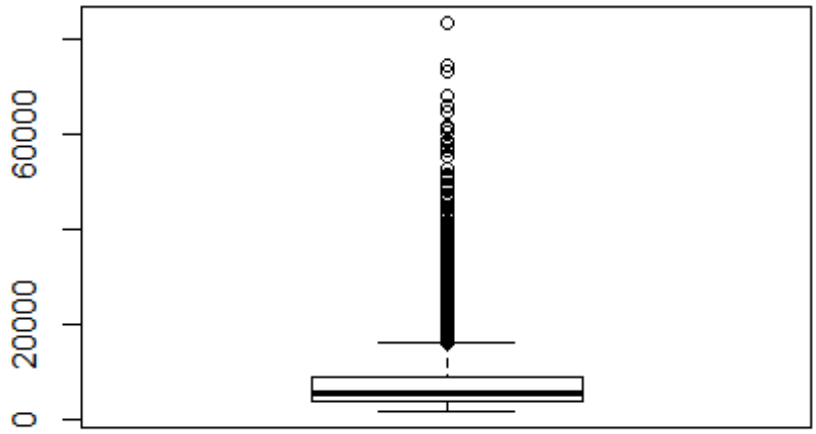
```
## [1] "Washington" "Arizona"    "Nevada"     "California" "Oregon"
```

```
table(df$State)
```

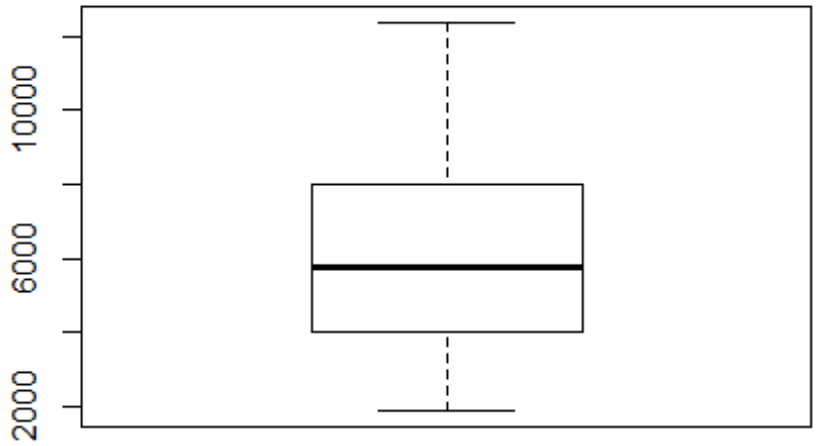
```
##
##    Arizona California     Nevada     Oregon Washington
##       1703       3150        882       2601        798
```

```
df$State = as.factor(df$State)
```

```r
#Customer Lifetime value
summary(df$`Customer Lifetime Value`)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1898    3994    5780    8005    8962   83325

boxplot(df$`Customer Lifetime Value`)
```




```r
df$`Customer Lifetime Value` = ifelse(df$`Customer Lifetime Value` >
15457,8005,
                                      df$`Customer Lifetime Value`)
summary(df$`Customer Lifetime Value`)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1898    3994    5780    6351    8005   15446

df$`Customer Lifetime Value` = ifelse(df$`Customer Lifetime Value` >
12367,6351,
                                      df$`Customer Lifetime Value`)
boxplot(df$`Customer Lifetime Value`)
```

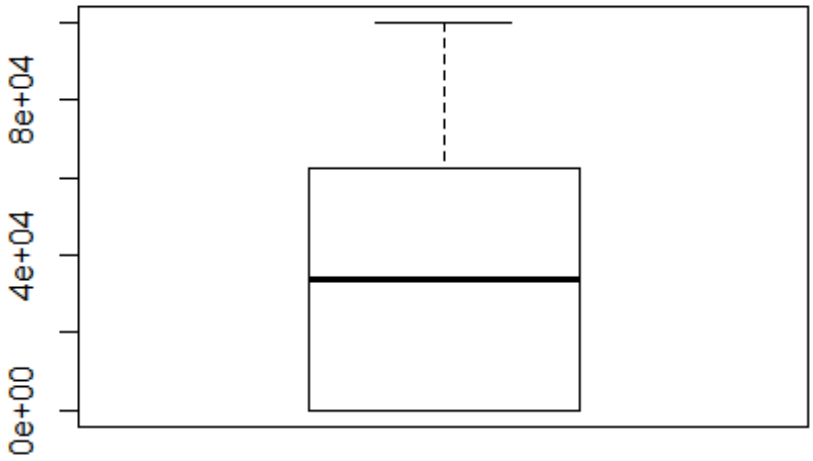



```
#Response
unique(df$Response)

## [1] "No"  "Yes"

df$Response = ifelse(df$Response == "Yes",1,0)

#Coverage
unique(df$Coverage)

## [1] "Basic"    "Extended" "Premium"

df$Coverage = as.factor(df$Coverage)

#Education
unique(df$Education)

## [1] "Bachelor"             "College"              "Master"
## [4] "High School or Below" "Doctor"

df$Education = factor(df$Education,
                      levels = c("High School or
Below","Bachelor","College","Master","Doctor"),
                      labels = c(0,1,1,1,2))
df$Education = as.numeric(df$Education)

For Schooling 0,
For Collage 1,
For Doctor 2
```

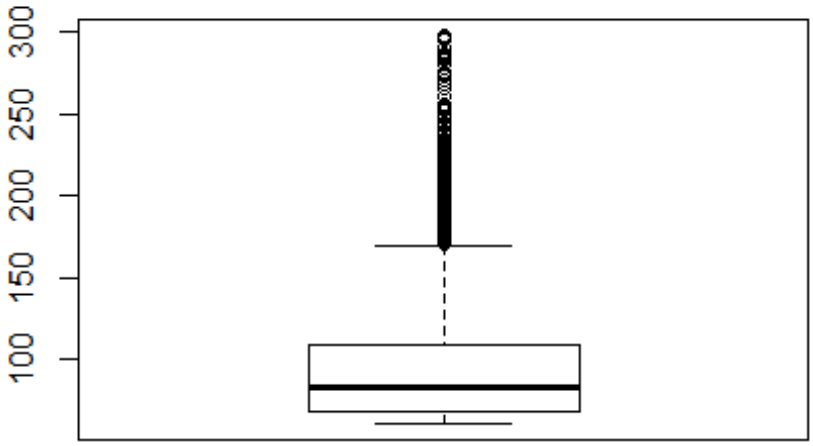
```r
#Effective TO Date
months = strftime(df$`Effective To Date`,"%m")
df$month = as.numeric(months)
days = strftime(df$`Effective To Date`,"%d")
df$day = as.numeric(days)
df$`Effective To Date` = NULL
```

Extrect months and days from date column

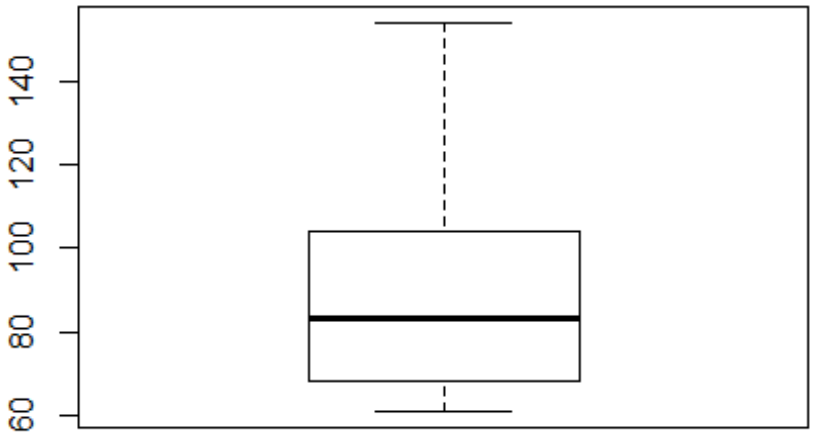
```r
#EmploymentStatus
unique(df$EmploymentStatus)

## [1] "Employed"      "Unemployed"    "Medical Leave" "Disabled"
## [5] "Retired"

df$EmploymentStatus = ifelse(df$EmploymentStatus == "Unemployed",0,1)

#Gender
unique(df$Gender)

## [1] "F" "M"

df$Gender  = ifelse(df$Gender == "M",1,0)

#INcome
summary(df$Income)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0   33890   37657   62320   99981

boxplot(df$Income)
```




```
#Location code
unique(df$`Location Code`)

## [1] "Suburban" "Rural"    "Urban"

#Marital Status
df$`Marital Status` = ifelse(df$`Marital Status` == "Married",1,0)

#Monthly Premium
summary(df$`Monthly Premium Auto`)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   61.00   68.00   83.00   93.22  109.00  298.00

boxplot(df$`Monthly Premium Auto`)
```



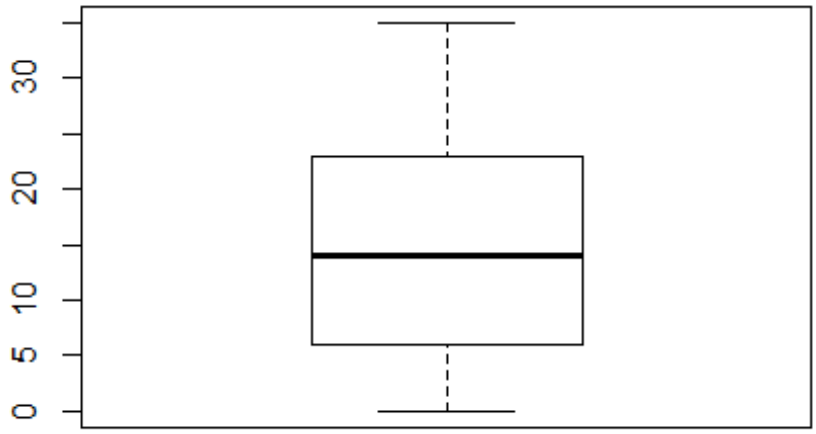

```
df$`Monthly Premium Auto` = ifelse(df$`Monthly Premium Auto` >
154,93,df$`Monthly Premium Auto`)
boxplot(df$`Monthly Premium Auto`)
```



```r
#months since last claim
summary(df$`Months Since Last Claim`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     6.0    14.0    15.1    23.0    35.0
```

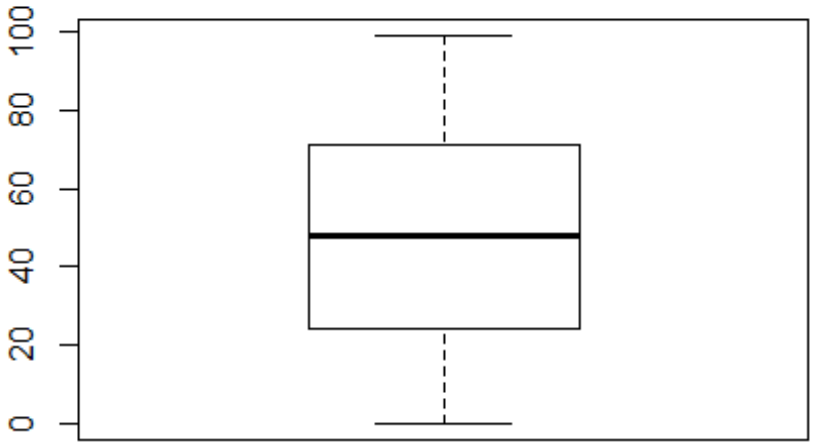
```r
boxplot(df$`Months Since Last Claim`)
```



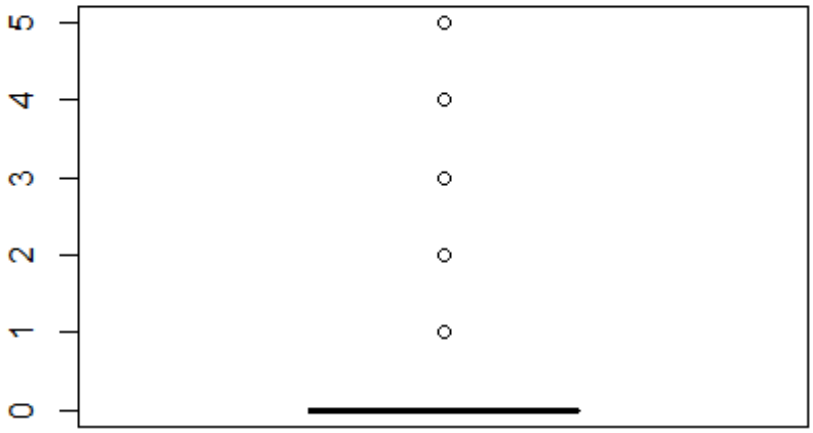

```r
#months since policy inception
summary(df$`Months Since Policy Inception`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   24.00   48.00   48.06   71.00   99.00
```

```r
boxplot(df$`Months Since Policy Inception`)
```



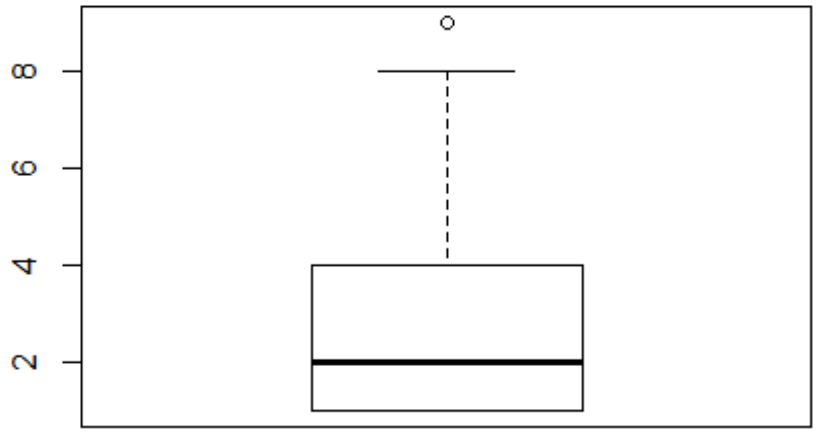

```
#number of open complaints
summary(df$`Number of Open Complaints`)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3844  0.0000  5.0000

boxplot(df$`Number of Open Complaints`)
```



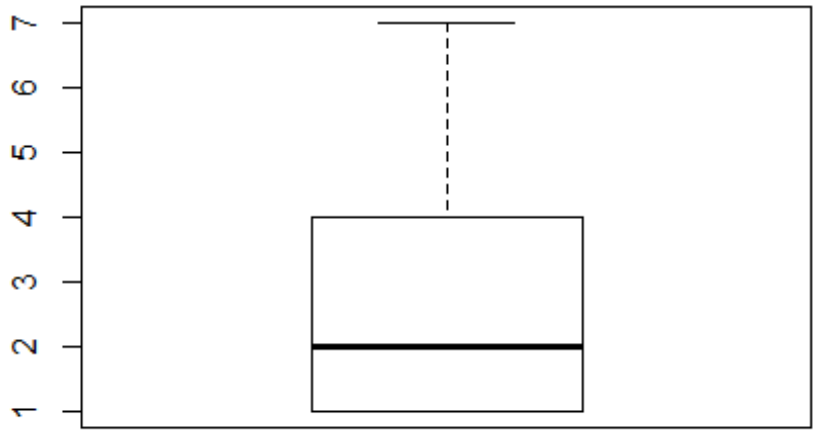

```
table(df$`Number of Open Complaints`)

##
##    0    1    2    3    4    5
## 7252 1011  374  292  149   56

df$`Number of Open Complaints` = NULL
```

This Variable is invariant

```
#number of policies
summary(df$`Number of Policies`)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.966   4.000   9.000

boxplot(df$`Number of Policies`)
```




```r
df$`Number of Policies` = ifelse(df$`Number of Policies` > 7.5,7,df$`Number of Policies`)
boxplot(df$`Number of Policies`)
```



```r
#Policy Type
table(df$`Policy Type`)
```

```
##
## Corporate Auto  Personal Auto   Special Auto
##           1968           6788            378
```

```r
library(stringr)
#Policy
table(df$Policy)
```

```
##
## Corporate L1 Corporate L2 Corporate L3  Personal L1  Personal L2
##          359          595         1014         1240         2122
##  Personal L3   Special L1   Special L2   Special L3
##         3426           66          164          148
```

```r
df$Policy = str_sub(df$Policy,-1,-1)

#Renew offer type
table(df$`Renew Offer Type`)
```

```
##
## Offer1 Offer2 Offer3 Offer4
##   3752   2926   1432   1024
```

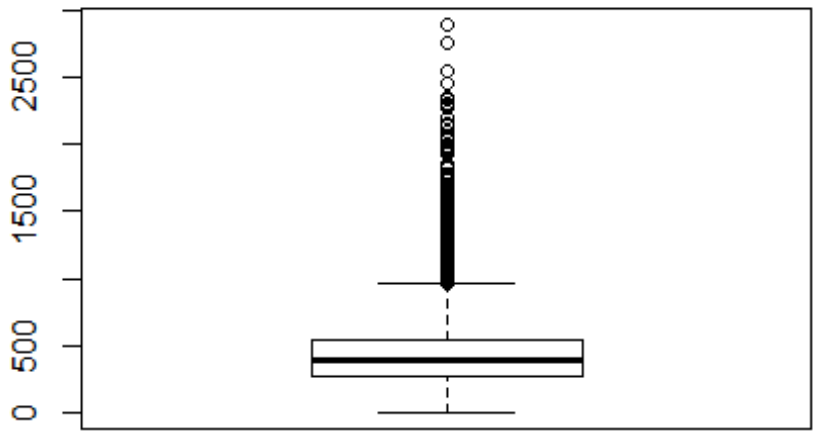
```r
df$`Renew Offer Type` = str_sub(df$`Renew Offer Type`,-1)

#Sales channel
table(df$`Sales Channel`)
```

```
##
##       Agent      Branch Call Center         Web
##        3477        2567        1765        1325
```

```r
#Total claim amount
summary(df$`Total Claim Amount`)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.099  272.258  383.945  434.089  547.515 2893.240
```

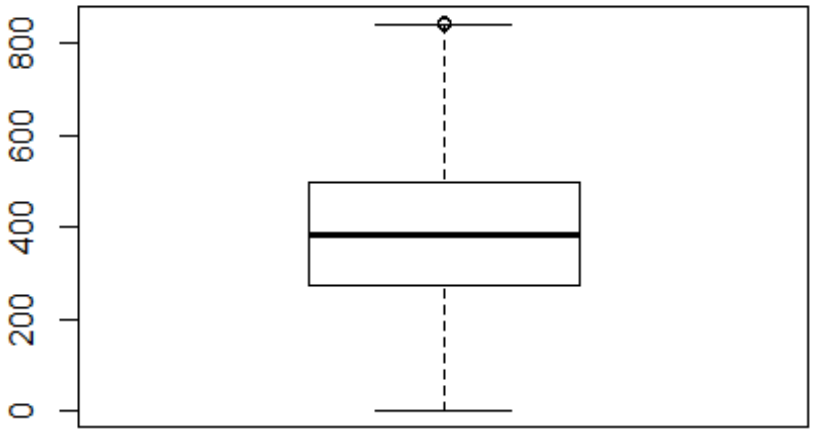
```r
boxplot(df$`Total Claim Amount`)
```




```
length(which(df$`Total Claim Amount` > 434 + (547.51-272.258)*1.5))
```

```
## [1] 693
```

```
df$`Total Claim Amount` = ifelse(df$`Total Claim Amount` > 847,434,df$`Total Claim Amount`)
boxplot(df$`Total Claim Amount`)
```




```
#Vehicle Class
table(df$`Vehicle Class`)

##
## Four-Door Car    Luxury Car    Luxury SUV    Sports Car           SUV
##          4621           163           184           484          1796
##  Two-Door Car
##          1886

df$`Vehicle Class` = factor(df$`Vehicle Class`,
                            levels = c("Luxury Car","Luxury SUV","Sports
Car","Four-Door Car","SUV","Two-Door Car"),
                            labels = c(3,3,3,2,2,1))
df$`Vehicle Class` = as.numeric(df$`Vehicle Class`)

#Vehicle size
table(df$`Vehicle Size`)

##
##   Large Medsize   Small
##     946    6424    1764

df$`Vehicle Size` = factor(df$`Vehicle Size`,
                           levels = c("Large","Medsize","Small"),
                           c(3,2,1))
df$`Vehicle Size` = as.numeric(df$`Vehicle Size`)
```

3. Modelling

```
df$`Number of Policies` = log(df$`Number of Policies`)
df$`Customer Lifetime Value` = log(df$`Customer Lifetime Value`)
df$`Monthly Premium Auto` = log(df$`Monthly Premium Auto`)
```

Dividing data into test and train

```
library(caTools)
set.seed(123)
split = sample.split(df$`Customer Lifetime Value`,SplitRatio = 0.8)
train = subset(df,split == TRUE)
test = subset(df,split == F)

#model 1
regg = lm(formula =  `Customer Lifetime Value` ~ . ,
          data = train[-1])
summary(regg)

##
## Call:
## lm(formula = `Customer Lifetime Value` ~ ., data = train[-1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5815 -0.2397 -0.1211  0.1536  1.0873
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    5.553e+00  1.101e-01  50.457  < 2e-16 ***
## StateCalifornia                1.239e-02  1.077e-02   1.150  0.25001
## StateNevada                    1.007e-02  1.481e-02   0.680  0.49629
## StateOregon                    2.019e-02  1.115e-02   1.810  0.07034 .
## StateWashington               -1.130e-02  1.535e-02  -0.736  0.46164
## Response                       2.514e-03  1.130e-02   0.222  0.82402
## CoverageExtended               2.466e-02  9.488e-03   2.599  0.00937 **
## CoveragePremium                9.689e-02  1.464e-02   6.619 3.88e-11 ***
## Education                     -1.608e-02  7.403e-03  -2.172  0.02990 *
## EmploymentStatus               6.093e-02  1.279e-02   4.765 1.92e-06 ***
## Gender                        -4.562e-03  7.517e-03  -0.607  0.54395
## Income                         4.895e-07  1.901e-07   2.575  0.01005 *
## `Location Code`Suburban       -2.664e-02  1.694e-02  -1.573  0.11587
## `Location Code`Urban          -1.353e-02  1.485e-02  -0.911  0.36235
## `Marital Status`               1.279e-02  8.072e-03   1.584  0.11322
## `Monthly Premium Auto`         6.528e-01  2.317e-02  28.171  < 2e-16 ***
## `Months Since Last Claim`     -3.160e-04  3.722e-04  -0.849  0.39592
## `Months Since Policy Inception` 6.167e-05  1.348e-04   0.457  0.64737
## `Number of Policies`           3.486e-01  5.333e-03  65.366  < 2e-16 ***
## `Policy Type`Personal Auto     3.808e-03  9.117e-03   0.418  0.67623
## `Policy Type`Special Auto      2.768e-02  2.035e-02   1.360  0.17377
## Policy2                       -1.363e-02  1.101e-02  -1.238  0.21570
```

```
## Policy3                          -8.253e-03  1.019e-02  -0.810  0.41784
## `Renew Offer Type`2              -4.885e-02  9.276e-03  -5.267 1.43e-07 ***
## `Renew Offer Type`3              -2.535e-02  1.137e-02  -2.229  0.02587 *
## `Renew Offer Type`4              -2.705e-02  1.320e-02  -2.050  0.04042 *
## `Sales Channel`Branch            -1.150e-03  9.345e-03  -0.123  0.90204
## `Sales Channel`Call Center       -1.697e-03  1.048e-02  -0.162  0.87134
## `Sales Channel`Web               -1.243e-02  1.172e-02  -1.060  0.28909
## `Total Claim Amount`              5.516e-05  3.789e-05   1.456  0.14552
## `Vehicle Class`                  -6.231e-02  7.573e-03  -8.228 2.23e-16 ***
## `Vehicle Size`                    5.376e-03  7.036e-03   0.764  0.44489
## month                            -1.843e-02  7.529e-03  -2.448  0.01440 *
## day                              -7.784e-04  4.344e-04  -1.792  0.07321 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3184 on 7273 degrees of freedom
## Multiple R-squared:  0.4932, Adjusted R-squared:  0.4909
## F-statistic: 214.5 on 33 and 7273 DF,  p-value: < 2.2e-16

#R-squared 0.491 and 0.03009
```

Remove Variables one-by-one to show p -value

```
#model2
regg = lm(formula =  `Customer Lifetime Value` ~ . ,
          data = train[-c(1,2,4,6,8,9,10,11,13,14,16,17,18,19,20,22,23,24)])
summary(regg)

##
## Call:
## lm(formula = `Customer Lifetime Value` ~ ., data = train[-c(1,
##     2, 4, 6, 8, 9, 10, 11, 13, 14, 16, 17, 18, 19, 20, 22, 23,
##     24)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6106 -0.2406 -0.1232  0.1574  1.0833
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              5.364405   0.094636  56.685  < 2e-16 ***
## CoverageExtended         0.021742   0.009470   2.296   0.0217 *
## CoveragePremium          0.098941   0.014624   6.766 1.43e-11 ***
## EmploymentStatus         0.082378   0.008559   9.624  < 2e-16 ***
## `Monthly Premium Auto`   0.679752   0.019960  34.056  < 2e-16 ***
## `Number of Policies`     0.351984   0.005282  66.641  < 2e-16 ***
## `Vehicle Class`         -0.062766   0.007571  -8.290  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3192 on 7300 degrees of freedom
```

```
## Multiple R-squared:  0.4888, Adjusted R-squared:  0.4884
## F-statistic:  1163 on 6 and 7300 DF,  p-value: < 2.2e-16

#R-squared 0.4888 and mape 0.03016

#Get the pridiction of fitted value
pred = predict(regg,newdata = test[-1])
test$pred = pred
Error = test$`Customer Lifetime Value` - test$pred

#Calculating MAPE
(sum((abs(test$`Customer Lifetime Value`-test$pred))/test$`Customer Lifetime
Value`))/nrow(test)

## [1] 0.03013484
```

Checking of Assumption

```
Residuals should be uncorrelated ##Autocorrelation
Null H0: residuals from a linear regression are uncorrelated. Value should be
close to 2.
Less than 1 and greater than 3 -> concern
Should get a high p value

library(car)

## Loading required package: carData

dwt(regg)

##  lag Autocorrelation D-W Statistic p-value
##    1    -0.005318895       2.01047   0.604
##  Alternative hypothesis: rho != 0

#Checking multicollinearity
vif(regg) # should be within 2. If it is greater than 10 then serious problem

##                            GVIF Df GVIF^(1/(2*Df))
## Coverage                1.443216  2        1.096056
## EmploymentStatus        1.000601  1        1.000301
## `Monthly Premium Auto` 1.628558  1        1.276150
## `Number of Policies`   1.000908  1        1.000454
## `Vehicle Class`        1.174774  1        1.083870
```

Heteroscedasticity

```
# Breusch-Pagan test
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(regg)  # Null hypothesis -> error is homogenious (p value should be
more than 0.05)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  regg
## BP = 810.54, df = 6, p-value < 2.2e-16
```

Normality testing Null hypothesis is data is normal.

```
resids = regg$residuals

library(nortest)
ad.test(resids)
```

```
##
##  Anderson-Darling normality test
##
## data:  resids
## A = 372.55, p-value < 2.2e-16
```

Anderson-Darling test for normality P- value is > 0.05