Fcon <- dbConnect(drv=RSQLite::SQLite(), dbname="pumssqlite")

setwd("C:\\analytics\_greatlakes\\data\_mining\\assignment\\csv\_hus")

dbWriteTable(conn = con,name="a",value = "ss13husa.csv",header=TRUE)

myquery <- dbSendQuery(con,"select \* from a limit 5")

dbClearResult(myquery)

dbWriteTable(conn = con,name="b",value = "ss13husb.csv",header=TRUE)

## THIS TABLE WILL HAVE ALL THE RECORDS

myquery <- dbSendQuery(con,"create table c as select \* from a union all select \* from b")

## Combining both the tables and create a new table with non null VALP records

myquery1 <- myquery <- dbSendQuery(con,"create table data as select \* from a where length(VALP)=0 union all select \* from b where length(VALP)=0")

> myquery1 <- myquery <- dbSendQuery(con,"select MAX(VALP),MIN(VALP),AVG(VALP),BLD FROM data GROUP BY BLD")

> dbFetch(myquery1)

MAX(VALP) MIN(VALP) AVG(VALP) BLD

1 4727000 110 58413.15 1

2 4775000 100 259255.39 2

3 4775000 100 253182.31 3

4 4775000 100 306625.84 4

5 4775000 110 274677.20 5

6 4775000 110 230417.31 6

7 4775000 110 236917.79 7

8 4775000 110 315591.58 8

9 4775000 110 408888.40 9

10 2738000 130 63769.06 10

> myquery1 <- myquery <- dbSendQuery(con,"select COUNT(SERIALNO) FROM data GROUP BY BLD")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

COUNT(SERIALNO)

1 57007

2 716293

3 44703

4 9855

5 6801

6 5705

7 4487

8 5160

9 8886

10 794

I CAN CLUB 1 AND 10. AND 2 TO 9 IN ANOTHER GROUP.

**FULP : FUEL PRICE**

**WE CAN REMOVE THIS AS THIS MEANS MORE LOGICAL FOR THE BLD=10**

> myquery1 <- dbSendQuery(con,"select MAX(FULP),MIN(FULP),AVG(FULP),BLD FROM data GROUP BY BLD ")

> dbFetch(myquery1)

MAX(FULP) MIN(FULP) AVG(FULP) BLD

1 1 118.728349 1

2 1 174.879350 2

3 1 92.562334 3

4 1 274.400406 4

5 1 146.558741 5

6 1 19.119018 6

7 1 8.923111 7

8 1 8.297093 8

9 1 7.793158 9

10 2100 1 45.847607 10

This fuel price can only be used for building = 10.

> myquery1 <- dbSendQuery(con,"select COUNT(SERIALNO) FROM DATA WHERE FULP = 2")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

COUNT(SERIALNO)

1 734696

IT MEANS FOR MOST OF THE HOUSES FULP IS INCLUDED WITH THE CONDO FEES

SEE THE BELOW PERCENTAGE

> myquery1 <- dbSendQuery(con,"SELECT A.BLD\_CNT,B.BLD\_FULP2,B.BLD\_FULP2\*100/A.BLD\_CNT PCT\_FULP2 FROM (select COUNT(SERIALNO)BLD\_CNT,BLD FROM DATA GROUP BY BLD) A,(SELECT COUNT(SERIALNO)BLD\_FULP2,BLD FROM DATA WHERE FULP=2 GROUP BY BLD) B WHERE A.BLD=B.BLD")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

BLD\_CNT BLD\_FULP2 PCT\_FULP2

1 57007 47786 83

2 716293 610594 85

3 44703 40926 91

4 9855 8097 82

5 6801 5817 85

6 5705 5132 89

7 4487 4042 90

8 5160 4444 86

9 8886 7160 80

10 794 698 87

SHOULD WE REMOVE GAS PRICE FROM THE PREDICTOR AS LOTS OF DATA HAS FREE GASP.

> myquery1 <- dbSendQuery(con,"SELECT A.FREE\_GASP,B.BLD\_CNT,(A.FREE\_GASP\*100/B.BLD\_CNT),A.BLD FROM (SELECT COUNT(SERIALNO)FREE\_GASP,BLD FROM DATA WHERE GASP=3 GROUP BY BLD)A,(SELECT BLD,COUNT(SERIALNO)BLD\_CNT FROM DATA GROUP BY BLD)B WHERE A.BLD = B.BLD")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

FREE\_GASP BLD\_CNT (A.FREE\_GASP\*100/B.BLD\_CNT) BLD

1 30799 57007 54 1

2 211245 716293 29 2

3 11927 44703 26 3

4 1416 9855 14 4

5 2015 6801 29 5

6 2361 5705 41 6

7 2095 4487 46 7

8 2536 5160 49 8

9 4214 8886 47 9

10 314 794 39 10

MAX HOUSE IS ALSO PRESENT WHERE NOT GAS PRICE IS THERE.

> myquery1 <- dbSendQuery(con,"SELECT AVG(VALP),MAX(VALP),MIN(VALP),BLD FROM DATA WHERE GASP >3 GROUP BY BLD ")

> dbFetch(myquery1)

AVG(VALP) MAX(VALP) MIN(VALP) BLD

1 57056.30 4727000 110 1

2 269215.67 4775000 100 2

3 273144.58 4775000 100 3

4 320909.88 4775000 110 4

5 309919.81 4775000 110 5

6 263375.52 4775000 110 6

7 281046.68 4775000 110 7

8 351887.09 4775000 120 8

9 478982.05 4775000 120 9

10 60250.87 2057000 150 10

> myquery1 <- dbSendQuery(con,"SELECT AVG(VALP),MAX(VALP),MIN(VALP),BLD FROM DATA WHERE GASP <3 GROUP BY BLD ")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

AVG(VALP) MAX(VALP) MIN(VALP) BLD

1 49349.34 2386000 150 1

2 320974.34 4775000 100 2

3 284729.14 4775000 110 3

4 327340.42 4775000 150 4

5 320075.68 4775000 110 5

6 277837.50 4775000 110 6

7 305642.63 4775000 120 7

8 442806.05 4775000 110 8

9 459462.71 4775000 110 9

10 75596.82 700000 130 10

HFL

We can see that the average price for the solar powered house is on higher side. Can we make a model without the solar house.

|  |
| --- |
| > myquery1 <- dbSendQuery(con,"SELECT AVG(VALP),MAX(VALP),MIN(VALP),COUNT(SERIALNO),HFL FROM DATA WHERE BLD BETWEEN 2 AND 9 GROUP BY HFL ")  Warning message:  Closing open result set, pending rows  > dbFetch(myquery1)  AVG(VALP) MAX(VALP) MIN(VALP) COUNT(SERIALNO) HFL  1 279449.0 4775000 100 410311 1  2 235384.6 4775000 110 56203 2  3 229849.1 4775000 100 229593 3  4 308106.0 4775000 110 53970 4  5 161956.7 3103000 130 1637 5  6 202039.0 4775000 110 28545 6  7 575509.0 4727000 130 672 7  8 275267.1 4775000 110 4236 8  9 363382.9 4775000 110 5613 9  10 237155.7 4775000 160 11110 0  Warning message:  In rsqlite\_fetch(res@ptr, n = n) :  Column `HFL`: mixed type, first seen values of type integer, coercing other values of type string  > myquery1 <- dbSendQuery(con,"SELECT AVG(VALP),MAX(VALP),MIN(VALP),COUNT(SERIALNO),HFL FROM  DATA WHERE BLD IN(1,10) GROUP BY HFL ")  Warning message:  Closing open result set, pending rows  > dbFetch(myquery1)  AVG(VALP) MAX(VALP) MIN(VALP) COUNT(SERIALNO) HFL  1 49909.28 4727000 110 11501 1  2 62274.72 2545000 110 8539 2  3 58490.69 4727000 110 29694 3  4 56621.44 4727000 140 2911 4  5 67572.73 280000 500 88 5  6 77602.44 2545000 120 2931 6  7 200833.33 1571000 7000 12 7  8 69599.24 1945000 230 485 8  9 65629.95 2738000 200 424 9  10 62320.02 2883000 200 1216 0  Warning message:  In rsqlite\_fetch(res@ptr, n = n) :  Column `HFL`: mixed type, first seen values of type integer, coercing other values of type string |
| We can create a model for solar and another for non solar house.  **INSP – INSURANCE PRICE**  **WE CAN SEE THAT THE FOR THE BLD TYPE= 10 THE INSURANCE PRICE IS HIGH. WE NEED TO CHECK IT**  **WHILE BUILDING THE MODEL.**  > myquery1 <- dbSendQuery(con,"SELECT count(SERIALNO),BLD,MIN(INSP),MAX(INSP),MIN(VALP),MAX(VALP) FROM DATA WHERE INSP>0 GROUP BY BLD")  Warning message:  Closing open result set, pending rows  > dbFetch(myquery1)  count(SERIALNO) BLD MIN(INSP) MAX(INSP) MIN(VALP) MAX(VALP)  1 33302 1 4 110 4727000  2 644462 2 4 100 4775000  3 38990 3 4 100 4775000  4 8773 4 4 110 4775000  5 5712 5 4 110 4775000  6 4603 6 4 110 4775000  7 3532 7 4 110 4775000  8 3927 8 4 120 4775000  9 6649 9 4 120 4775000  10 293 10 4 5200 300 2057000  MHP  > myquery1 <- dbSendQuery(con,"SELECT COUNT(SERIALNO),BLD FROM DATA WHERE FMHP=0 GROUP BY BLD")  Warning message:  Closing open result set, pending rows  > dbFetch(myquery1)  COUNT(SERIALNO) BLD  1 44549 1  2 716293 2  3 44703 3  4 9855 4  5 6801 5  6 5705 6  7 4487 7  8 5160 8  9 8886 9  10 794 10  > myquery1 <- dbSendQuery(con,"SELECT COUNT(SERIALNO),BLD FROM DATA WHERE FMHP=1 GROUP BY BLD")  Warning message:  Closing open result set, pending rows  > dbFetch(myquery1)  COUNT(SERIALNO) BLD  1 12458 1  WE CAN EXCLUDE THIS. AS THIS IS APPLICABLE ONLY FOR BLD=1  MRGP  > myquery1 <- dbSendQuery(con,"SELECT COUNT(SERIALNO),FMRGP FROM DATA GROUP BY FMRGP")  Warning message:  Closing open result set, pending rows  > dbFetch(myquery1)  COUNT(SERIALNO) FMRGP  1 802151 0  2 57540 1  WE CAN EXCLUDE ALL MORTGAGE RELATED COLUMNS AS FRACTIONAL DATA SHOWS HOUSE IS UNDER  MORTGAGE.  RMSP  WE CAN EXCLUDE RMSP. AS IT SEEMS NO RELATION.  > myquery1 <- dbSendQuery(con,"SELECT MAX(X.VALP),Y.RMSP FROM DATA X ,(SELECT DISTINCT RMSP FROM DATA) Y WHERE X.RMSP=Y.RMSP GROUP BY Y.RMSP")  Warning message:  Closing open result set, pending rows  > dbFetch(myquery1)  MAX(X.VALP) RMSP  1 4775000 1  2 4775000 2  3 4775000 3  4 4775000 4  5 4775000 5  6 4775000 6  7 4775000 7  8 4775000 8  9 4775000 9  10 4775000 10  11 4775000 11  12 4775000 12  13 4775000 13  14 3934000 14  15 2738000 15  16 2880000 16  17 2221000 17  18 4532000 18  19 3103000 19  20 4727000 20  21 4775000 21  22 2738000 22  23 3934000 24  24 3972000 25  WATP CAN BE INCLUDED.  > dbFetch(myquery1)  COUNT(\*) BLD MAX(VALP) MAX(WATP) MIN(WATP)  1 50418 1 4727000 2  2 713472 2 4775000 2  3 38245 3 4775000 2  4 9258 4 4775000 2  5 4475 5 4775000 2  6 2522 6 4775000 2  7 1551 7 4775000 2  8 1392 8 4775000 2  9 2144 9 4775000 2  10 483 10 2738000 3400 2 |
| |  | | --- | |  | |

YBL

> myquery1 <- dbSendQuery(con,"SELECT YBL,MIN(VALP),MAX(VALP),COUNT(SERIALNO),BLD FROM DATA GROUP BY YBL,BLD ORDER BY BLD,YBL ")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

YBL MIN(VALP) MAX(VALP) COUNT(SERIALNO) BLD

1 1 200 4727000 247 1

2 2 400 4727000 186 1

3 3 140 4727000 663 1

4 4 140 2883000 3072 1

5 5 110 2880000 11709 1

6 6 110 2313000 12944 1

7 7 110 2880000 17655 1

8 8 110 2880000 6173 1

9 9 110 3103000 950 1

10 10 110 1250000 888 1

11 11 130 1250000 657 1

12 12 180 2552000 564 1

13 13 110 500000 384 1

14 14 110 1826000 380 1

15 15 1000 815000 239 1

16 16 130 1826000 224 1

17 17 500 389000 72 1

18 1 110 4775000 100246 2

19 2 110 4775000 41662 2

20 3 100 4775000 95270 2

21 4 100 4775000 84064 2

22 5 100 4775000 104046 2

23 6 110 4775000 83778 2

24 7 100 4775000 96692 2

25 8 110 4775000 55783 2

26 9 110 4775000 12771 2

27 10 110 4775000 12310 2

28 11 110 4775000 9033 2

29 12 120 4775000 6434 2

30 13 110 4775000 4502 2

31 14 110 4775000 4285 2

32 15 110 4775000 2862 2

33 16 110 4775000 2018 2

34 17 230 4727000 537 2

35 1 110 4775000 5904 3

36 2 110 4775000 2012 3

37 3 100 4775000 2715 3

38 4 120 4775000 2402 3

39 5 110 4775000 6314 3

40 6 100 4775000 9119 3

41 7 110 4775000 6823 3

42 8 110 4727000 4368 3

43 9 110 4775000 1232 3

44 10 120 2738000 1104 3

45 11 130 2552000 903 3

46 12 140 2738000 624 3

47 13 130 2057000 418 3

48 14 160 2133000 354 3

49 15 230 2738000 211 3

50 16 170 1300000 160 3

51 17 130000 1826000 40 3

52 1 120 4775000 4872 4

53 2 110 4775000 890 4

54 3 130 4775000 1044 4

55 4 120 4775000 867 4

56 5 130 3224000 697 4

57 6 100 4727000 544 4

58 7 120 4775000 432 4

59 8 180 2057000 255 4

60 9 370 950000 67 4

61 10 1000 4532000 59 4

62 11 9600 1500000 37 4

63 12 200 625000 31 4

64 13 550 900000 21 4

65 14 200 730000 18 4

66 15 80000 1603000 13 4

67 16 40000 1400000 6 4

68 17 170000 830000 2 4

69 1 110 4775000 1798 5

70 2 200 3103000 301 5

71 3 160 4727000 379 5

72 4 250 3972000 484 5

73 5 110 2500000 1125 5

74 6 110 3972000 1230 5

75 7 120 2880000 680 5

76 8 120 3972000 397 5

77 9 200 700000 96 5

78 10 300 1000000 95 5

79 11 130 2200000 96 5

80 12 1000 3224000 48 5

81 13 280 1000000 25 5

82 14 25000 2313000 24 5

83 15 1000 400000 15 5

84 16 150000 1800000 7 5

85 17 200000 200000 1 5

86 1 160 4775000 554 6

87 2 280 4727000 126 6

88 3 130 2782000 192 6

89 4 120 4775000 461 6

90 5 110 1500000 1132 6

91 6 110 2880000 1540 6

92 7 110 4775000 844 6

93 8 110 2880000 407 6

94 9 130 2880000 97 6

95 10 110 2880000 91 6

96 11 170 1000000 95 6

97 12 140 1100000 60 6

98 13 180 1000000 48 6

99 14 60000 1300000 25 6

100 15 75000 2000000 21 6

101 16 160000 670000 11 6

102 17 800 800 1 6

103 1 190 4775000 362 7

104 2 130 750000 94 7

105 3 130 2880000 192 7

106 4 130 4727000 433 7

107 5 120 3972000 927 7

108 6 110 2880000 1111 7

109 7 120 2000000 642 7

110 8 180 2880000 320 7

111 9 110 4775000 102 7

112 10 130 2880000 93 7

113 11 110 950000 68 7

114 12 20000 950000 57 7

115 13 45000 800000 38 7

116 14 210 900000 33 7

117 15 159000 4775000 9 7

118 16 130000 575000 5 7

119 17 430000 430000 1 7

120 1 270 4775000 576 8

121 2 200 4775000 128 8

122 3 140 4775000 291 8

123 4 150 4775000 561 8

124 5 110 4775000 1186 8

125 6 120 4775000 1006 8

126 7 120 2880000 488 8

127 8 130 2880000 382 8

128 9 200 2880000 117 8

129 10 120 2000000 137 8

130 11 300 2880000 109 8

131 12 290 3972000 81 8

132 13 260 1600000 43 8

133 14 99000 2300000 31 8

134 15 7400 2880000 16 8

135 16 350000 1000000 6 8

136 17 360000 400000 2 8

137 1 170 4775000 760 9

138 2 120 4775000 283 9

139 3 110 4775000 689 9

140 4 110 4775000 1430 9

141 5 120 4775000 1870 9

142 6 120 4775000 1356 9

143 7 160 4775000 678 9

144 8 150 4775000 624 9

145 9 200 4775000 231 9

146 10 230 4775000 251 9

147 11 150 4775000 278 9

148 12 280 4775000 214 9

149 13 950 3103000 127 9

150 14 300 2300000 63 9

151 15 500 4727000 22 9

152 16 8500 1000000 10 9

153 1 1000 1945000 35 10

154 2 4000 100000 8 10

155 3 3000 600000 14 10

156 4 500 500000 25 10

157 5 200 900000 80 10

158 6 150 2738000 156 10

159 7 200 2000000 206 10

160 8 500 530000 117 10

161 9 1000 357000 31 10

162 10 2500 100000 38 10

163 11 2000 150000 19 10

164 12 1200 600000 14 10

165 13 130 95000 11 10

166 14 4000 450000 11 10

167 15 6000 2057000 14 10

168 16 22000 175000 9 10

169 17 6500 390000 6 10

We can see that upto a certain age the house price is stable but then its decreasing.

WKEXREL

THIS CAN BE REMOVED AS 20% DATA IS MISSING.

> myquery1 <- dbSendQuery(con,"SELECT COUNT(\*),WKEXREL FROM DATA WHERE LENGTH(WKEXREL) > 0")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

COUNT(\*) WKEXREL

1 620857 7

> myquery1 <- dbSendQuery(con,"SELECT COUNT(\*),WKEXREL FROM DATA WHERE LENGTH(WKEXREL) <= 0")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

COUNT(\*) WKEXREL

1 238834

> myquery1 <- dbSendQuery(con,"SELECT COUNT(\*),WKEXREL FROM DATA WHERE LENGTH(WKEXREL) <= 0 AND BLD=2")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

COUNT(\*) WKEXREL

1 176945

> myquery1 <- dbSendQuery(con,"SELECT COUNT(\*) FROM DATA")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

COUNT(\*)

1 859691

FIBEROP

AS WE ARE SEEING THAT AVERAGE PRICE IS HIGHER WHERE FIBEROPTICS IS PRESENT. WE CAN TAKE THIS AS A FACTOR PARAM.

> myquery1 <- myquery <- dbSendQuery(con,"SELECT COUNT(\*),FIBEROP,MAX(VALP),MIN(VALP),AVG(VALP) FROM DATA GROUP BY FIBEROP")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

COUNT(\*) FIBEROP MAX(VALP) MIN(VALP) AVG(VALP)

1 79496 1 4775000 110 346319.8

2 592592 2 4775000 100 261912.4

3 187603 0 4775000 100 160938.1

**HHL**

> myquery1 <- myquery <- dbSendQuery(con,"SELECT COUNT(\*),MAX(VALP),MIN(VALP),AVG(VALP),HHL FROM DATA WHERE BLD=2 GROUP BY HHL")

Warning message:

Closing open result set, pending rows

> dbFetch(myquery1)

COUNT(\*) MAX(VALP) MIN(VALP) AVG(VALP) HHL

1 603303 4775000 100 250443.9 1

2 52139 4775000 110 237619.3 2

3 26809 4775000 110 379750.1 3

4 19780 4775000 110 428797.1 4

5 5338 4775000 110 288011.4 5

6 8924 4775000 160 226389.7 0

Warning message:

In rsqlite\_fetch(res@ptr, n = n) :

Column `HHL`: mixed type, first seen values of type integer, coercing other values of type string

> myquery <- dbSendQuery(con,"CREATE TABLE DATA\_B2

+ AS SELECT SERIALNO,VALP,ACR,AGS,BDSP,ELEP,INSP,WATP,YBL,FINCP,MV,HINCP,SMOCP,TAXP,WIF

+ FROM DATA

+ where

+ BLD=2

+ AND HFL <> 7

+ AND FINCP > 0

+ AND HHL = 1

+ AND LNGI =1

+ AND HINCP >0

+ AND LENGTH(TAXP) > 0 ")

Warning message:

Closing open result set, pending rows

> View(DATA\_B2)

Error in View : object 'DATA\_B2' not found

> myquery1 <- myquery <- dbSendQuery(con,"select \* from DATA\_B2")

Warning message:

Closing open result set, pending rows

> final <- dbFetch(myquery1)

# Define a function to calculate the missing data

pMiss <- function(x){sum(is.na(x))/length(x)\*100}

miceplot <- aggr(final, col = c("navyblue","yellow"),numbers=TRUE,gap = 2)

#Train-validation split

train <- final %>% dplyr::sample\_frac(.70)

> test <- dplyr::anti\_join(final,train,by="SERIALNO")

> nrow(train)

[1] 419052

> nrow(test)

[1] 179593

Install.packages(“biglm”)

Library(biglm)

ml\_formula <- formula(VALP ~ ACR+AGS+BDSP+ELEP+INSP+WATP+YBL+FINCP+MV+HINCP+SMOCP+TAXP+WIF)

# Train a logistic regression model

(ml\_log <- ml\_logistic\_regression(train, ml\_formula))

Including the Factor Variable

> myquery1 <- dbSendQuery(con,"CREATE TABLE FINAL\_2 AS SELECT SERIALNO,VALP,ACR,BDSP,ELEP,INSP,WATP,YBL,FINCP,MV,HINCP,SMOCP,TAXP,WIF,CASE WHEN KIT =1 THEN 1 ELSE 0 END AS KIT\_1,CASE WHEN LNGI =1 THEN 1 ELSE 0 END AS LNGI\_1,CASE WHEN HHT = 1 THEN 1 ELSE 0 END AS HHT\_1,CASE WHEN HHT = 2 THEN 1 ELSE 0 END AS HHT\_2,

+ CASE WHEN HHT = 3 THEN 1 ELSE 0 END AS HHT\_3,CASE WHEN HHT = 4 THEN 1 ELSE 0 END AS HHT\_4,CASE WHEN HHT = 5 THEN 1 ELSE 0 END AS HHT\_5,CASE WHEN HHT = 6 THEN 1 ELSE 0 END AS HHT\_6,R18,CASE WHEN R60 = 1 THEN 1 ELSE 0 END AS R\_601,CASE WHEN R60 =2 THEN 1 ELSE 0 END AS R\_602,CASE WHEN R65 = 1 THEN 1 ELSE 0 END AS R\_651,CASE WHEN R65 =2 THEN 1 ELSE 0 END AS R\_652,

+ FACCESSP,

+ FBATHP,

+ FBROADBNDP,

+ FFIBEROPP

+ FROM DATA

+ where

+ BLD=2

+ AND HFL <> 7

+ AND FINCP > 0

+ AND HHL = 1

+ AND LNGI =1

+ AND HINCP >0

+ AND LENGTH(TAXP) > 0")

Warning message:

Closing open result set, pending rows

> myquery1 <- dbSendQuery(con,"SELECT \* FROM FINAL\_2")

Warning message:

Closing open result set, pending rows

> final2 <- dbFetch(myquery1)

> str(final2)

'data.frame': 598645 obs. of 30 variables:

$ SERIALNO : int 154 156 286 312 316 447 776 944 1117 1242 ...

$ ACR : int 1 1 1 1 2 2 2 1 1 1 ...

$ BDSP : int 3 3 4 5 3 3 2 2 4 2 ...

$ ELEP : int 350 300 100 240 130 80 190 50 150 70 ...

$ INSP : int 350 980 50 2200 1700 1200 600 900 1200 850 ...

$ WATP : int 480 700 370 50 250 50 600 2 630 460 ...

$ YBL : int 2 5 2 8 2 4 6 3 6 4 ...

$ FINCP : int 151000 0 0 136000 52600 81600 35000 0 175000 73000 ...

$ MV : int 6 7 7 4 7 7 6 6 6 4 ...

$ HINCP : int 151000 39930 5400 136000 52600 81600 35000 1600 175000 73000 ...

$ SMOCP : int 426 926 522 1444 346 857 294 142 1508 678 ...

$ TAXP : int 3 6 3 26 5 10 2 5 24 9 ...

$ WIF : int 2 0 0 2 0 2 2 0 2 1 ...

$ KIT\_1 : int 1 1 1 1 1 1 1 1 1 1 ...

$ LNGI\_1 : int 1 1 1 1 1 1 1 1 1 1 ...

$ HHT\_1 : int 1 0 0 1 1 1 1 0 1 1 ...

$ HHT\_2 : int 0 0 0 0 0 0 0 0 0 0 ...

$ HHT\_3 : int 0 0 0 0 0 0 0 0 0 0 ...

$ HHT\_4 : int 0 0 0 0 0 0 0 0 0 0 ...

$ HHT\_5 : int 0 0 0 0 0 0 0 0 0 0 ...

$ HHT\_6 : int 0 1 1 0 0 0 0 1 0 0 ...

$ R18 : int 0 0 0 0 0 0 0 0 0 0 ...

$ R\_601 : int 0 1 1 0 0 1 0 1 0 1 ...

$ R\_602 : int 0 0 0 0 1 0 0 0 0 0 ...

$ R\_651 : int 0 0 1 0 0 1 0 1 0 0 ...

$ R\_652 : int 0 0 0 0 1 0 0 0 0 0 ...

$ FACCESSP : int 0 0 0 0 0 0 0 1 1 0 ...

$ FBATHP : int 0 0 0 1 0 0 0 0 0 0 ...

$ FBROADBNDP: int 0 0 1 1 1 0 0 0 0 0 ...

$ FFIBEROPP : int 0 0 1 0 0 0 0 0 0 0 ...

CREATE TABLE FINAL\_2 AS SELECT SERIALNO,ACR,AGS,BDSP,ELEP,INSP,WATP,YBL,FINCP,MV,HINCP,SMOCP,TAXP,WIF,CASE WHEN KIT =1 THEN 1 ELSE 0 END AS KIT\_1,CASE WHEN LNGI =1 THEN 1 ELSE 0 END AS LNGI\_1,CASE WHEN HHT = 1 THEN 1 ELSE 0 END AS HHT\_1,CASE WHEN HHT = 2 THEN 1 ELSE 0 END AS HHT\_2,

CASE WHEN HHT = 3 THEN 1 ELSE 0 END AS HHT\_3,CASE WHEN HHT = 4 THEN 1 ELSE 0 END AS HHT\_4,CASE WHEN HHT = 5 THEN 1 ELSE 0 END AS HHT\_5,CASE WHEN HHT = 6 THEN 1 ELSE 0 END AS HHT\_6,R18,CASE WHEN R60 = 1 THEN 1 ELSE 0 END AS R\_601,CASE WHEN R60 =2 THEN 1 ELSE 0 END AS R\_602,CASE WHEN R65 = 1 THEN 1 ELSE 0 END AS R\_651,CASE WHEN R65 =2 THEN 1 ELSE 0 END AS R\_652,

FACCESSP,

FBATHP,

FBROADBANDP,

FBIBEROPP

FROM DATA

where

BLD=2

AND HFL <> 7

AND FINCP > 0

AND HHL = 1

AND LNGI =1

AND HINCP >0

AND LENGTH(TAXP) > 0

#Train-validation split

train <- final2 %>% dplyr::sample\_frac(.70)

test <- dplyr::anti\_join(final\_2,train,by="SERIALNO")

str(train)

test(train)

ml\_formula <- formula(VALP ~ ACR + BDSP+ELEP+INSP+WATP+ FINCP +MV+HINCP+SMOCP+TAXP+WIF+ KIT\_1+HHT\_1+HHT\_2 + HHT\_3+HHT\_4+HHT\_5+HHT\_6+R18+ R\_601+R\_602+R\_651+R\_652+FBROADBNDP+FFIBEROPP)

model\_lm <- lm(ml\_formula, train)

summary(model\_lm)