House Loan Data Challenge

Paul Zuo

Sep 23, 2017

EDA

Data Import

Load the data into R.

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
require(rpart)
```

```
## Loading required package: rpart
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(randomForest)
```

```
## Loading required package: randomForest
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
setwd("/Users/paulzuo/Documents/Penn 2017-2018/Data_Challenges/waf")
data1 <- read.csv("datasource1.csv", header = TRUE,
                  stringsAsFactors = F, na.strings = "")
data2 <- read.csv("datasource2.csv", header = TRUE,
                  stringsAsFactors = F, na.strings = "")
data3 <- read.csv("datasource3.csv", header = TRUE,
                  stringsAsFactors = F, na.strings = "")
data4 <- read.csv("datasource4.csv", header = TRUE,
                  stringsAsFactors = F, na.strings = "")
```

Let’s explore each of these datasets.

```
head(data1)
```

```
##   LOANID ORIG_TERM   BAL_DIFF NOTE_RATE LOAN_PURP ORIGINAL_FICO_SCORE
## 1 380247       360  -25948.52      5.87     PURCH                 766
## 2    810       360  -20691.67      9.62      <NA>                 561
## 3   4860       180 -118047.94      5.50     SREFI                 712
## 4  52646       360  -15370.95      6.37     CREFI                 785
## 5   9443       300  -42255.94      6.37     SREFI                 766
## 6  22249       360  -60698.20      5.62     CREFI                 767
##   CURRENT_FICO_SCORE FICO.DIFF PROP_ZIP PROP_TYPE Loan.Issued
## 1                809        43    11235     CONDO           1
## 2                774       213    11211      <NA>           1
## 3                763        51    11229     TWO-4           1
## 4                816        31    11229     TWO-4           1
## 5                801        35    11235        SF           1
## 6                789        22    11214     CONDH           1
```

```
head(data2)
```

```
##   PROP_ZIP PROPERTY_TURNOVER LISTING_COUNT MEDIAN_PRICE MEDIAN_PPSQFT
## 1    11235             6.454           416       549000           491
## 2    11211             4.779            87      1097500           435
## 3    11229             4.276           237       568000           407
## 4    11235             6.454           416       549000           491
## 5    11214             4.737            91       585000           463
## 6    11206             4.080            28       568000           435
##   FORECLOSURERATIO  ZRI ZRI_YOY   ZHVI NEGATIVEEQUITY DELINQUENCY ZHVI_YOY
## 1            1.204 2177   0.067 447500          0.091       0.081    0.698
## 2            0.000 2921   0.047 961600          0.056       0.000    6.020
## 3            0.773 2143   0.070 555300          0.058       0.081    7.700
## 4            1.204 2177   0.067 447500          0.091       0.081    0.698
## 5            1.630 2154   0.107 551800          0.000       0.000    5.426
## 6            2.276 2654   0.056 480000          0.000       0.000    7.000
```

```
head(data3)
```

```
##   PROP_ZIP POPULATION_YOUTH POPULATION_ADULT POPULATION_ELDER
## 1    10001              877             7773             4295
## 2    10002            10884            43810            21151
## 3    10003            20682            82080            39630
## 4    10004              641             3354             1551
## 5    10005             2553            11359             4510
## 6    10006             1239             8372             3485
##   POPULATION_POVERTY POPULATION_EMPLOYED POPULATION_UNEMPLOYED
## 1              12885            64.00000              9.400000
## 2              71576            51.96667             10.640000
## 3             141552            55.95714              8.739286
## 4               5451            70.60000              5.350000
## 5              18271            77.35000              4.250000
## 6              12697            33.97500              2.550000
##   HOUSEHOLD_TOTAL HOUSEHOLD_NONFAMILY HOUSEHOLD_FAMILY
## 1            8259                6267             1992
## 2           31972               16914            15058
## 3           59322               28692            30630
## 4            2966                1984              982
## 5           10026                6522             3504
## 6            6876                4753             2123
##   HOUSEHOLD_MEDIANINCOME HOUSEHOLD_EXPEND_HOUSEHOLD HOUSEHOLD_MEDIANRENT
## 1                64151.5                    10366.0               1121.5
## 2                52604.9                     8323.4                823.5
## 3                44304.0                     7980.8                764.4
## 4               123525.0                    15783.5               1376.0
## 5               132186.5                    16798.5               1437.0
## 6                42181.3                    13402.4               1239.5
```

```
head(data4)
```

```
##   PROP_ZIP ROBBERY BURGLARY FELONY.ASSAULT GRAND.LARCENY MURDER RAPE
## 1    10001      12       13             16            73      0    0
## 2    10002      78       47            107           358      1   14
## 3    10003     119       66            192           406      3    0
## 4    10004       9       10              8           109      1    0
## 5    10005      17       41             18           306      0    0
## 6    10006      12       15             18           107      0    0
##   GRAND.LARCENY.OF.MOTOR.VEHICLE ALLFELONIES
## 1                              2         116
## 2                             32         637
## 3                             41         827
## 4                              2         139
## 5                              6         388
## 6                              2         154
```

In the second dataframe, we want to average out some of the features based on the zip code.

```
data2 <- data2 %>%
  group_by(PROP_ZIP) %>%
  summarise_each(funs(mean(., na.rm=TRUE)))
```

```
## `summarise_each()` is deprecated.
## Use `summarise_all()`, `summarise_at()` or `summarise_if()` instead.
## To map `funs` over all variables, use `summarise_all()`
```

Each of the dataframes has a variable for the property zip code. We will merge the data together by the property zip code. Let's first examine missing data.

```
sapply(data1, function(x) sum(is.na(x)))
```

```
##              LOANID           ORIG_TERM            BAL_DIFF
##                   0                   0                   0
##           NOTE_RATE           LOAN_PURP ORIGINAL_FICO_SCORE
##                   0                  17                   0
##  CURRENT_FICO_SCORE           FICO.DIFF            PROP_ZIP
##                   0                   0                   0
##           PROP_TYPE         Loan.Issued
##                  27                   0
```

```
sapply(data2, function(x) sum(is.na(x)))
```

```
##          PROP_ZIP PROPERTY_TURNOVER     LISTING_COUNT      MEDIAN_PRICE
##                 0                 0                 0                 0
##      MEDIAN_PPSQFT  FORECLOSURERATIO               ZRI           ZRI_YOY
##                 0                 0                 0                 0
##              ZHVI    NEGATIVEEQUITY       DELINQUENCY          ZHVI_YOY
##                 0                 0                 0                 0
```

```
sapply(data3, function(x) sum(is.na(x)))
```

```
##                 PROP_ZIP           POPULATION_YOUTH
##                        0                          0
##         POPULATION_ADULT           POPULATION_ELDER
##                        0                          0
##       POPULATION_POVERTY        POPULATION_EMPLOYED
##                        0                          0
##    POPULATION_UNEMPLOYED            HOUSEHOLD_TOTAL
##                        0                          0
##      HOUSEHOLD_NONFAMILY           HOUSEHOLD_FAMILY
##                        0                          0
##    HOUSEHOLD_MEDIANINCOME HOUSEHOLD_EXPEND_HOUSEHOLD
##                        0                          0
##      HOUSEHOLD_MEDIANRENT
##                        0
```

```
sapply(data4, function(x) sum(is.na(x)))
```

```
##                         PROP_ZIP                         ROBBERY
##                                0                               0
##                         BURGLARY                  FELONY.ASSAULT
##                                0                               0
##                    GRAND.LARCENY                          MURDER
##                                0                               0
##                             RAPE GRAND.LARCENY.OF.MOTOR.VEHICLE
##                                0                               0
##                      ALLFELONIES
##                                0
```

```
## let's drop all the rows of data frame 1 where the loan purpose and property type are
n't given...
data1 <- na.omit(data1)
```

The greater concern behind having several rows with missing values is that the data could've either been entered wrong or the software tool to scrape the data could be faulty. In either case, we want to exclude the rows entirely. Wrong data is worrisome and can be an indicator of some bug in the logging code. Therefore, I would like to talk to the software engineer who implemented the code to see if, perhaps, there are some bugs which affect the data significantly. Now, we merge the dataframes.

```
unique(data2$PROP_ZIP) ## see that there's more not just necessarily one of each zip cod
e
```

```
##   [1] 10001 10002 10003 10004 10005 10006 10007 10009 10010 10011 10012
##  [12] 10013 10014 10016 10017 10018 10019 10021 10022 10023 10024 10025
##  [23] 10026 10027 10028 10029 10031 10032 10033 10034 10035 10036 10037
##  [34] 10038 10039 10040 10044 10065 10069 10075 10128 10280 10303 10304
##  [45] 10305 10306 10308 10309 10310 10312 10314 10453 10455 10456 10458
##  [56] 10459 10460 10461 10462 10463 10464 10465 10466 10467 10469 10470
##  [67] 10471 10472 10473 10474 10475 11004 11101 11102 11103 11104 11105
##  [78] 11106 11109 11201 11203 11205 11206 11208 11209 11210 11211 11212
##  [89] 11213 11214 11215 11219 11220 11222 11228 11229 11230 11233 11234
## [100] 11235 11238 11354 11355 11356 11357 11358 11360 11361 11362 11363
## [111] 11364 11365 11366 11367 11368 11369 11370 11372 11373 11374 11375
## [122] 11377 11378 11379 11385 11411 11412 11413 11414 11415 11416 11417
## [133] 11418 11419 11420 11421 11422 11423 11426 11427 11428 11429 11432
## [144] 11433 11434 11435 11436 11691 11692 11694
```

```
data <- merge(x = data1, y = data2, by = "PROP_ZIP", all.x = TRUE)
data <- merge(x = data, y = data3, by = "PROP_ZIP", all.x = TRUE)
data <- merge(x = data, y = data4, by = "PROP_ZIP", all.x = TRUE)
```

Data Cleaning

Explore the row structure

```
head(data)
```

```
##   PROP_ZIP LOANID ORIG_TERM  BAL_DIFF NOTE_RATE LOAN_PURP
## 1    10001 280342       180 -12767.58      3.87     RREFI
## 2    10001 322242       360  -4637.34      3.87     PURCH
## 3    10002 249146       360  -4802.52      4.75     PURCH
## 4    10002 264268       360  -8501.26      4.87     PURCH
## 5    10002 267022       360  -7706.00      4.75     PURCH
## 6    10002 306152       360  -1889.77      4.50     RREFI
##   ORIGINAL_FICO_SCORE CURRENT_FICO_SCORE FICO.DIFF PROP_TYPE Loan.Issued
## 1                 720                746        26      COOP           1
## 2                 715                715         0     CONDH           1
## 3                 811                818         7      COOP           0
## 4                 770                791        21      COOP           1
## 5                 772                785        13      COOP           0
## 6                 783                774        -9     CONDH           0
##   PROPERTY_TURNOVER LISTING_COUNT MEDIAN_PRICE MEDIAN_PPSQFT
## 1             8.002            59       568000          1663
## 2             8.002            59       568000          1663
## 3             5.010           144       568000          1332
## 4             5.010           144       568000          1332
## 5             5.010           144       568000          1332
## 6             5.010           144       568000          1332
##   FORECLOSURERATIO  ZRI ZRI_YOY   ZHVI NEGATIVEEQUITY DELINQUENCY ZHVI_YOY
## 1                1 3900  -0.019 480000              0           0        7
## 2                1 3900  -0.019 480000              0           0        7
## 3                0 3739  -0.028 480000              0           0        7
## 4                0 3739  -0.028 480000              0           0        7
## 5                0 3739  -0.028 480000              0           0        7
## 6                0 3739  -0.028 480000              0           0        7
##   POPULATION_YOUTH POPULATION_ADULT POPULATION_ELDER POPULATION_POVERTY
## 1              877             7773             4295              12885
## 2              877             7773             4295              12885
## 3            10884            43810            21151              71576
## 4            10884            43810            21151              71576
## 5            10884            43810            21151              71576
## 6            10884            43810            21151              71576
##   POPULATION_EMPLOYED POPULATION_UNEMPLOYED HOUSEHOLD_TOTAL
## 1            64.00000                  9.40            8259
## 2            64.00000                  9.40            8259
## 3            51.96667                 10.64           31972
## 4            51.96667                 10.64           31972
## 5            51.96667                 10.64           31972
## 6            51.96667                 10.64           31972
##   HOUSEHOLD_NONFAMILY HOUSEHOLD_FAMILY HOUSEHOLD_MEDIANINCOME
## 1                6267             1992                64151.5
## 2                6267             1992                64151.5
## 3               16914            15058                52604.9
## 4               16914            15058                52604.9
## 5               16914            15058                52604.9
## 6               16914            15058                52604.9
##   HOUSEHOLD_EXPEND_HOUSEHOLD HOUSEHOLD_MEDIANRENT ROBBERY BURGLARY
## 1                    10366.0               1121.5      12       13
## 2                    10366.0               1121.5      12       13
## 3                     8323.4                823.5      78       47
```

```
## 4                    8323.4                   823.5      78       47
## 5                    8323.4                   823.5      78       47
## 6                    8323.4                   823.5      78       47
##   FELONY.ASSAULT GRAND.LARCENY MURDER RAPE GRAND.LARCENY.OF.MOTOR.VEHICLE
## 1             16            73      0    0                              2
## 2             16            73      0    0                              2
## 3            107           358      1   14                             32
## 4            107           358      1   14                             32
## 5            107           358      1   14                             32
## 6            107           358      1   14                             32
##   ALLFELONIES
## 1         116
## 2         116
## 3         637
## 4         637
## 5         637
## 6         637
```

Look at the correlation between the features.

```
str(data)
```

```
## 'data.frame':    1344 obs. of  42 variables:
##  $ PROP_ZIP                    : int  10001 10001 10002 10002 10002 10002 10002 100
02 10002 10002 ...
##  $ LOANID                      : int  280342 322242 249146 264268 267022 306152 236
913 288058 239627 217358 ...
##  $ ORIG_TERM                   : int  180 360 360 360 360 360 360 180 360 360 ...
##  $ BAL_DIFF                    : num  -12768 -4637 -4803 -8501 -7706 ...
##  $ NOTE_RATE                   : num  3.87 3.87 4.75 4.87 4.75 4.5 4.25 3.25 4.25
 4.37 ...
##  $ LOAN_PURP                   : chr  "RREFI" "PURCH" "PURCH" "PURCH" ...
##  $ ORIGINAL_FICO_SCORE         : int  720 715 811 770 772 783 760 782 802 808 ...
##  $ CURRENT_FICO_SCORE          : int  746 715 818 791 785 774 793 788 809 807 ...
##  $ FICO.DIFF                   : int  26 0 7 21 13 -9 33 6 7 -1 ...
##  $ PROP_TYPE                   : chr  "COOP" "CONDH" "COOP" "COOP" ...
##  $ Loan.Issued                 : int  1 1 0 1 0 0 0 1 0 1 ...
##  $ PROPERTY_TURNOVER           : num  8 8 5.01 5.01 5.01 ...
##  $ LISTING_COUNT               : num  59 59 144 144 144 144 144 144 144 144 ...
##  $ MEDIAN_PRICE                : num  568000 568000 568000 568000 568000 568000 568
000 568000 568000 568000 ...
##  $ MEDIAN_PPSQFT               : num  1663 1663 1332 1332 1332 ...
##  $ FORECLOSURERATIO            : num  1 1 0 0 0 0 0 0 0 0 ...
##  $ ZRI                         : num  3900 3900 3739 3739 3739 ...
##  $ ZRI_YOY                     : num  -0.019 -0.019 -0.028 -0.028 -0.028 -0.028 -0.
028 -0.028 -0.028 -0.028 ...
##  $ ZHVI                        : num  480000 480000 480000 480000 480000 480000 480
000 480000 480000 480000 ...
##  $ NEGATIVEEQUITY              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ DELINQUENCY                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ZHVI_YOY                    : num  7 7 7 7 7 7 7 7 7 7 ...
##  $ POPULATION_YOUTH            : int  877 877 10884 10884 10884 10884 10884 10884 1
0884 10884 ...
##  $ POPULATION_ADULT            : int  7773 7773 43810 43810 43810 43810 43810 43810
43810 43810 ...
##  $ POPULATION_ELDER            : int  4295 4295 21151 21151 21151 21151 21151 21151
21151 21151 ...
##  $ POPULATION_POVERTY          : int  12885 12885 71576 71576 71576 71576 71576 715
76 71576 71576 ...
##  $ POPULATION_EMPLOYED         : num  64 64 52 52 52 ...
##  $ POPULATION_UNEMPLOYED       : num  9.4 9.4 10.6 10.6 10.6 ...
##  $ HOUSEHOLD_TOTAL             : int  8259 8259 31972 31972 31972 31972 31972 31972
31972 31972 ...
##  $ HOUSEHOLD_NONFAMILY         : int  6267 6267 16914 16914 16914 16914 16914 16914
16914 16914 ...
##  $ HOUSEHOLD_FAMILY            : int  1992 1992 15058 15058 15058 15058 15058 15058
15058 15058 ...
##  $ HOUSEHOLD_MEDIANINCOME      : num  64152 64152 52605 52605 52605 ...
##  $ HOUSEHOLD_EXPEND_HOUSEHOLD  : num  10366 10366 8323 8323 8323 ...
##  $ HOUSEHOLD_MEDIANRENT        : num  1122 1122 824 824 824 ...
##  $ ROBBERY                     : int  12 12 78 78 78 78 78 78 78 78 ...
##  $ BURGLARY                    : int  13 13 47 47 47 47 47 47 47 47 ...
##  $ FELONY.ASSAULT              : int  16 16 107 107 107 107 107 107 107 107 ...
##  $ GRAND.LARCENY               : int  73 73 358 358 358 358 358 358 358 358 ...
##  $ MURDER                      : int  0 0 1 1 1 1 1 1 1 1 ...
```

```
##  $ RAPE                          : int  0 0 14 14 14 14 14 14 14 14 ...
##  $ GRAND.LARCENY.OF.MOTOR.VEHICLE: int  2 2 32 32 32 32 32 32 32 32 ...
##  $ ALLFELONIES                   : int  116 116 637 637 637 637 637 637 637 637 ...
```

```
summary(data)
```

```
##     PROP_ZIP         LOANID         ORIG_TERM        BAL_DIFF
##  Min.   :10001   Min.   :   226   Min.   :120.0   Min.   :-272448
##  1st Qu.:10025   1st Qu.:232988   1st Qu.:360.0   1st Qu.: -19496
##  Median :10463   Median :287552   Median :360.0   Median :  -9315
##  Mean   :10659   Mean   :265108   Mean   :332.6   Mean   : -17333
##  3rd Qu.:11370   3rd Qu.:326612   3rd Qu.:360.0   3rd Qu.:  -4357
##  Max.   :11694   Max.   :400887   Max.   :480.0   Max.   :   9686
##    NOTE_RATE      LOAN_PURP          ORIGINAL_FICO_SCORE CURRENT_FICO_SCORE
##  Min.   :2.000   Length:1344        Min.   :562.0       Min.   :448.0
##  1st Qu.:3.870   Class :character   1st Qu.:736.0       1st Qu.:725.0
##  Median :4.500   Mode  :character   Median :766.0       Median :774.0
##  Mean   :4.533                      Mean   :755.9       Mean   :750.2
##  3rd Qu.:5.000                      3rd Qu.:785.0       3rd Qu.:794.0
##  Max.   :8.000                      Max.   :820.0       Max.   :818.0
##    FICO.DIFF          PROP_TYPE          Loan.Issued     PROPERTY_TURNOVER
##  Min.   :-280.000   Length:1344        Min.   :0.0000   Min.   : 2.056
##  1st Qu.: -20.000   Class :character   1st Qu.:1.0000   1st Qu.: 4.102
##  Median :   1.500   Mode  :character   Median :1.0000   Median : 5.240
##  Mean   :  -5.691                      Mean   :0.9152   Mean   : 5.340
##  3rd Qu.:  24.000                      3rd Qu.:1.0000   3rd Qu.: 6.388
##  Max.   : 161.000                      Max.   :1.0000   Max.   :13.914
##  LISTING_COUNT    MEDIAN_PRICE     MEDIAN_PPSQFT    FORECLOSURERATIO
##  Min.   : 13.0   Min.   : 125000   Min.   : 160.0   Min.   :  0.000
##  1st Qu.: 87.0   1st Qu.: 469000   1st Qu.: 369.0   1st Qu.:  0.368
##  Median :144.0   Median : 568000   Median : 435.0   Median :  1.000
##  Mean   :151.1   Mean   : 762654   Mean   : 561.8   Mean   :  2.439
##  3rd Qu.:175.0   3rd Qu.: 808888   3rd Qu.: 497.0   3rd Qu.:  2.085
##  Max.   :462.0   Max.   :2775000   Max.   :1980.0   Max.   :294.118
##       ZRI           ZRI_YOY               ZHVI         NEGATIVEEQUITY
##  Min.   :1582   Min.   :-0.06600   Min.   : 108200   Min.   :0.00000
##  1st Qu.:2162   1st Qu.:-0.00700   1st Qu.: 449700   1st Qu.:0.00000
##  Median :2418   Median : 0.04100   Median : 480000   Median :0.04300
##  Mean   :2886   Mean   : 0.03869   Mean   : 575787   Mean   :0.05185
##  3rd Qu.:3710   3rd Qu.: 0.07600   3rd Qu.: 587300   3rd Qu.:0.09100
##  Max.   :6852   Max.   : 0.20200   Max.   :3028000   Max.   :0.25300
##   DELINQUENCY         ZHVI_YOY       POPULATION_YOUTH POPULATION_ADULT
##  Min.   :0.0000   Min.   :-12.954   Min.   :    0    Min.   :     0
##  1st Qu.:0.0000   1st Qu.:  6.377   1st Qu.: 5940    1st Qu.: 17018
##  Median :0.0000   Median :  7.000   Median :10999    Median : 42035
##  Mean   :0.0624   Mean   :  6.378   Mean   :15317    Mean   : 53934
##  3rd Qu.:0.1150   3rd Qu.:  8.060   3rd Qu.:20682    3rd Qu.: 72589
##  Max.   :0.3570   Max.   : 19.810   Max.   :62023    Max.   :209863
##  POPULATION_ELDER POPULATION_POVERTY POPULATION_EMPLOYED
##  Min.   :     0   Min.   :     0     Min.   : 0.00
##  1st Qu.:  7619   1st Qu.: 30256     1st Qu.:54.16
##  Median : 16650   Median : 69665     Median :57.61
##  Mean   : 26419   Mean   : 93826     Mean   :58.34
##  3rd Qu.: 34130   3rd Qu.:115511     3rd Qu.:64.05
##  Max.   :112661   Max.   :376671     Max.   :78.00
##  POPULATION_UNEMPLOYED HOUSEHOLD_TOTAL  HOUSEHOLD_NONFAMILY
##  Min.   : 0.000        Min.   :     0   Min.   :     0
##  1st Qu.: 6.157        1st Qu.: 10413   1st Qu.:  3496
##  Median : 8.739        Median : 29744   Median : 11297
```

```
##  Mean   : 9.065        Mean   : 43310   Mean   : 23242
##  3rd Qu.:11.204        3rd Qu.: 59322   3rd Qu.: 28692
##  Max.   :34.100        Max.   :185418   Max.   :104185
##  HOUSEHOLD_FAMILY HOUSEHOLD_MEDIANINCOME HOUSEHOLD_EXPEND_HOUSEHOLD
##  Min.   :    0    Min.   :     0         Min.   : 4740
##  1st Qu.: 7132    1st Qu.: 50809         1st Qu.: 7348
##  Median :13949    Median : 61691         Median : 8323
##  Mean   :20068    Mean   : 72988         Mean   :10358
##  3rd Qu.:28220    3rd Qu.:101718         3rd Qu.:14358
##  Max.   :81233    Max.   :155865         Max.   :19388
##  HOUSEHOLD_MEDIANRENT    ROBBERY          BURGLARY      FELONY.ASSAULT
##  Min.   :   0.0       Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 811.3       1st Qu.: 28.00   1st Qu.: 34.00   1st Qu.: 35.75
##  Median :1018.3       Median : 58.00   Median : 61.00   Median : 69.00
##  Mean   :1080.3       Mean   : 72.49   Mean   : 71.94   Mean   : 84.47
##  3rd Qu.:1425.0       3rd Qu.:115.00   3rd Qu.: 97.00   3rd Qu.:109.00
##  Max.   :2752.0       Max.   :216.00   Max.   :273.00   Max.   :328.00
##  GRAND.LARCENY        MURDER           RAPE
##  Min.   :   2.0   Min.   :0.000   Min.   : 0.000
##  1st Qu.:  80.0   1st Qu.:0.000   1st Qu.: 0.000
##  Median : 211.0   Median :1.000   Median : 0.000
##  Mean   : 361.7   Mean   :1.186   Mean   : 5.889
##  3rd Qu.: 544.0   3rd Qu.:2.000   3rd Qu.:12.000
##  Max.   :2105.0   Max.   :6.000   Max.   :29.000
##  GRAND.LARCENY.OF.MOTOR.VEHICLE  ALLFELONIES
##  Min.   :  0.00                 Min.   :   9.0
##  1st Qu.: 17.00                 1st Qu.: 235.0
##  Median : 30.00                 Median : 503.0
##  Mean   : 34.04                 Mean   : 631.7
##  3rd Qu.: 48.00                 3rd Qu.: 852.0
##  Max.   :111.00                 Max.   :2890.0
```

```
myvars <- names(data) %in% c("LOAN_PURP", "PROP_TYPE")
newdata <- data[!myvars]
data$LOAN_PURP <- as.factor(data$LOAN_PURP)
data$PROP_TYPE <- as.factor(data$PROP_TYPE)
data$Loan.Issued <- as.factor(data$Loan.Issued)
```

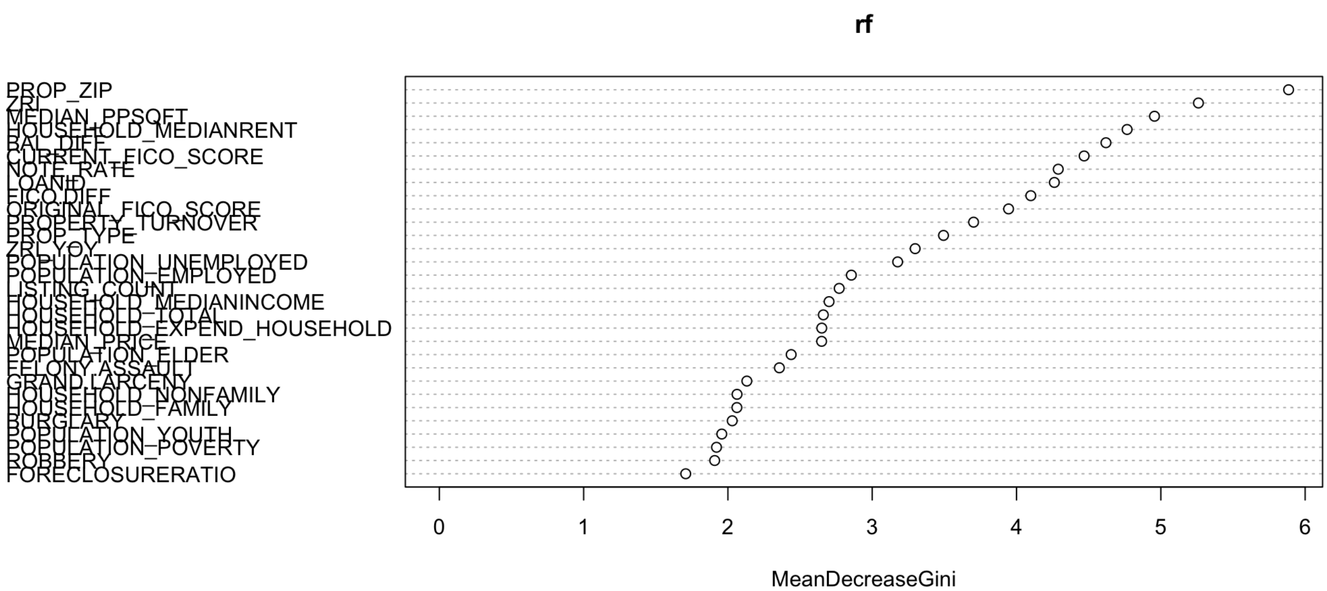
Now let’s take a look at the summary.

We notice that the loan issued rate is 90.7%- that is 90.7% of the people in the dataset got a loan issued. That’s pretty high.

```
train_sample = sample(nrow(data), size = nrow(data) * 0.66)
train_data = data[train_sample,]
test_data = data[-train_sample,]

rf = randomForest(y = train_data$Loan.Issued, x = train_data[, -11], ytest = test_data$L
oan.Issued, xtest = test_data[, -11], ntree = 100, mtry = 3, keep.forest = TRUE)

rf
```

```
##
## Call:
##  randomForest(x = train_data[, -11], y = train_data$Loan.Issued,      xtest = test_da
ta[, -11], ytest = test_data$Loan.Issued,      ntree = 100, mtry = 3, keep.forest = TRU
E)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 5.41%
## Confusion matrix:
##    0   1 class.error
## 0 47  28  0.37333333
## 1 20 792  0.02463054
##                 Test set error rate: 6.78%
## Confusion matrix:
##    0   1 class.error
## 0 17  22   0.5641026
## 1  9 409   0.0215311
```

```
varImpPlot(rf,type=2)
```




```
rf$importance
```

```
##                                  MeanDecreaseGini
## PROP_ZIP                                5.8854509
## LOANID                                  4.2621478
## ORIG_TERM                               1.4987895
## BAL_DIFF                                4.6196608
## NOTE_RATE                               4.2891164
## LOAN_PURP                               1.6538797
## ORIGINAL_FICO_SCORE                     3.9449965
## CURRENT_FICO_SCORE                      4.4684546
## FICO.DIFF                               4.0989237
## PROP_TYPE                               3.4939636
## PROPERTY_TURNOVER                       3.7024429
## LISTING_COUNT                           2.7709859
## MEDIAN_PRICE                            2.6493404
## MEDIAN_PPSQFT                           4.9561396
## FORECLOSURERATIO                        1.7075005
## ZRI                                     5.2606092
## ZRI_YOY                                 3.2974574
## ZHVI                                    1.0261165
## NEGATIVEEQUITY                          1.1399254
## DELINQUENCY                             1.4458496
## ZHVI_YOY                                0.7267032
## POPULATION_YOUTH                        1.9573832
## POPULATION_ADULT                        1.3188878
## POPULATION_ELDER                        2.4381943
## POPULATION_POVERTY                      1.9208200
## POPULATION_EMPLOYED                     2.8552335
## POPULATION_UNEMPLOYED                   3.1761640
## HOUSEHOLD_TOTAL                         2.6611222
## HOUSEHOLD_NONFAMILY                     2.0630602
## HOUSEHOLD_FAMILY                        2.0620350
## HOUSEHOLD_MEDIANINCOME                  2.7009406
## HOUSEHOLD_EXPEND_HOUSEHOLD              2.6503248
## HOUSEHOLD_MEDIANRENT                    4.7662528
## ROBBERY                                 1.9082000
## BURGLARY                                2.0298203
## FELONY.ASSAULT                          2.3564360
## GRAND.LARCENY                           2.1311320
## MURDER                                  0.5924100
## RAPE                                    0.8989414
## GRAND.LARCENY.OF.MOTOR.VEHICLE          1.7017548
## ALLFELONIES                             1.3633731
```

Here, we see that 7 of the most important features are balance difference (from the ending point - starting point), current FICO score, median PP SQFT, ZRI, employed population for the zip code, household median income for the zip code, and household median rent for the zip code. Let’s reassess the dataset, using just these features.

```
myvars <- names(data) %in% c("BAL_DIFF", "CURRENT_FICO_SCORE", "ZRI_YOY", "HOUSEHOLD_MED
IANINCOME", "HOUSEHOLD_MEDIANRENT", "MEDIAN_PPSQFT", "POPULATION_EMPLOYED", "Loan.Issue
d")
slimdata <- data[myvars]
slimdata$Loan.Issued <- as.factor(slimdata$Loan.Issued)
summary(slimdata)
```

```
##     BAL_DIFF       CURRENT_FICO_SCORE Loan.Issued MEDIAN_PPSQFT
##  Min.   :-272448   Min.   :448.0      0: 114      Min.   : 160.0
##  1st Qu.: -19496   1st Qu.:725.0      1:1230      1st Qu.: 369.0
##  Median :  -9315   Median :774.0                  Median : 435.0
##  Mean   : -17333   Mean   :750.2                  Mean   : 561.8
##  3rd Qu.:  -4357   3rd Qu.:794.0                  3rd Qu.: 497.0
##  Max.   :   9686   Max.   :818.0                  Max.   :1980.0
##     ZRI_YOY          POPULATION_EMPLOYED HOUSEHOLD_MEDIANINCOME
##  Min.   :-0.06600   Min.   : 0.00       Min.   :     0
##  1st Qu.:-0.00700   1st Qu.:54.16       1st Qu.: 50809
##  Median : 0.04100   Median :57.61       Median : 61691
##  Mean   : 0.03869   Mean   :58.34       Mean   : 72988
##  3rd Qu.: 0.07600   3rd Qu.:64.05       3rd Qu.:101718
##  Max.   : 0.20200   Max.   :78.00       Max.   :155865
##  HOUSEHOLD_MEDIANRENT
##  Min.   :   0.0
##  1st Qu.: 811.3
##  Median :1018.3
##  Mean   :1080.3
##  3rd Qu.:1425.0
##  Max.   :2752.0
```

We notice a few things:

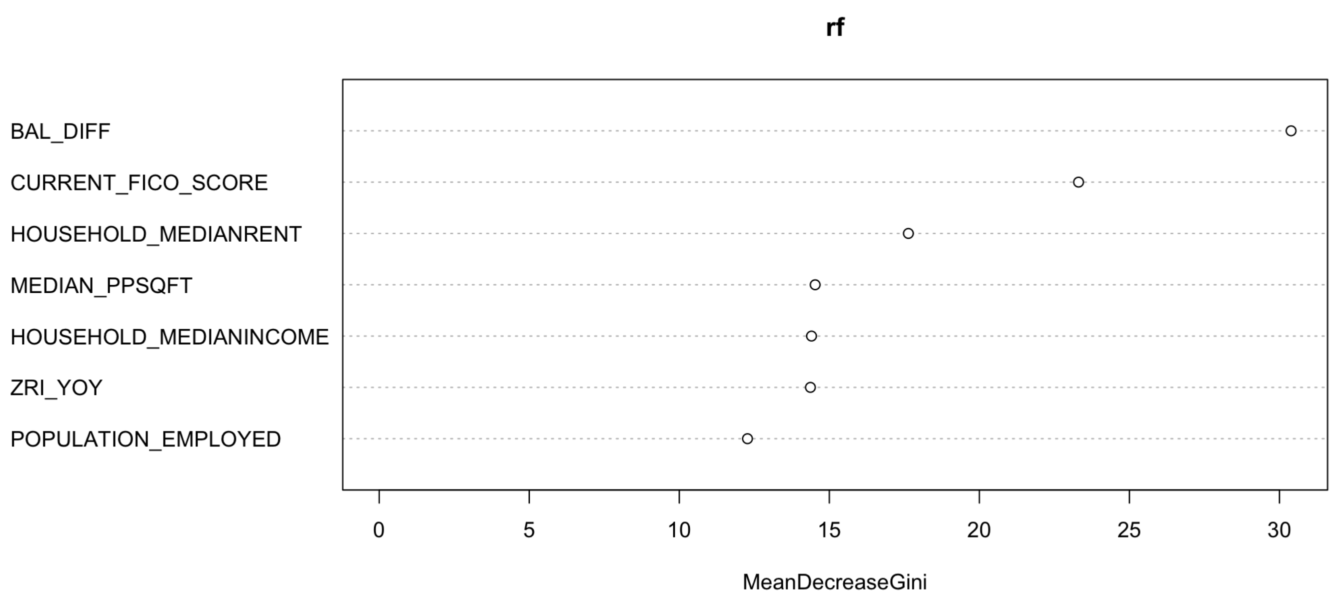
1. The sample of people in this dataset have a pretty large negative balance difference- the difference between the current loan balance and the original loan balance is fairly negative.
2. The median FICO score is 774, which indicates that most people in this dataset have at least a solid credit score (assuming a “very good” credit FICO score is 740-799)
3. The median PP SQ FT is $435, which indicates that the people looking to get a loan issued in this dataset are seeking out property that is much more expensive than the national average ($123 per sq foot). Makes sense, as most of the data comes from the greater NY city area.
4. The median ZRI_YOY is 0.041 which indicates that housing prices in general for this dataset are increasing year to year. It appears that there are rather few people looking for loans for properties in areas where the ZRI is decreasing year to year.
5. The Household median income is $72,988 on average, compared to the US national average $59,039 and the NYC average of $50,711.
6. the average household rent is $1080.3, compared to the NYC average household rent of $3,185.

Machine Learning

```
train_sample = sample(nrow(slimdata), size = nrow(slimdata) * 0.66)
train_data = slimdata[train_sample,]
test_data = slimdata[-train_sample,]

rf = randomForest(y = train_data$Loan.Issued, x = train_data[, -3], ytest = test_data$Lo
an.Issued, xtest = test_data[, -3], ntree = 100, mtry = 3, keep.forest = TRUE)

rf
```

```
##
## Call:
##  randomForest(x = train_data[, -3], y = train_data$Loan.Issued,      xtest = test_dat
a[, -3], ytest = test_data$Loan.Issued, ntree = 100,      mtry = 3, keep.forest = TRUE)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 6.88%
## Confusion matrix:
##    0   1 class.error
## 0 38  38  0.50000000
## 1 23 788  0.02836005
##                 Test set error rate: 6.35%
## Confusion matrix:
##    0   1 class.error
## 0 18  20  0.52631579
## 1  9 410  0.02147971
```

```
varImpPlot(rf,type=2)
```




```
rf$importance
```

```
##                          MeanDecreaseGini
## BAL_DIFF                         30.38769
## CURRENT_FICO_SCORE               23.30572
## MEDIAN_PPSQFT                    14.52822
## ZRI_YOY                          14.36997
## POPULATION_EMPLOYED              12.27273
## HOUSEHOLD_MEDIANINCOME           14.40616
## HOUSEHOLD_MEDIANRENT             17.63449
```

Recall that the sensitivity can be calculated as TP / (TP + FN) and the specificity can be calculated as TN / (FP + TN). Sensitivity measures the proportion of conversions that are correctly identified as such. Specificity, on the other hand, measures the ability to identify people who don't have a condition.

First, we note that the OOB error rate from the training set and the test set error rate are roughly the same (OOB error rate of 7.33% vs test error rate of 6.35%). This means that there isn't a huge amount of overfitting. The error rate is relatively low. But since only around 9.3% of the data points got loans issued, this is not that impressive- we started from a 93.65% accuracy if we predict everyone as loan issued. While 95.4% test accuracy is good, it isn't that shocking at the same time. Indeed around 50% of non-loan issues are predicted as loan issues.

If we cared about the very best possible accuracy or specifically minimizing false positive/negative, we would also use ROCR and find the best cut-off point. Since that isn't necessarily relevant here, we are fine with the 0.5 default cut off value used internally by random forest to make the prediction.

From the variable importance plot, we see that the balance difference feature is the most important feature by a decent margin.

So, let's rebuild the RF. Since the class for conversion is heavily unbalanced, let's change the weight a bit so that we do get some classified as 0.

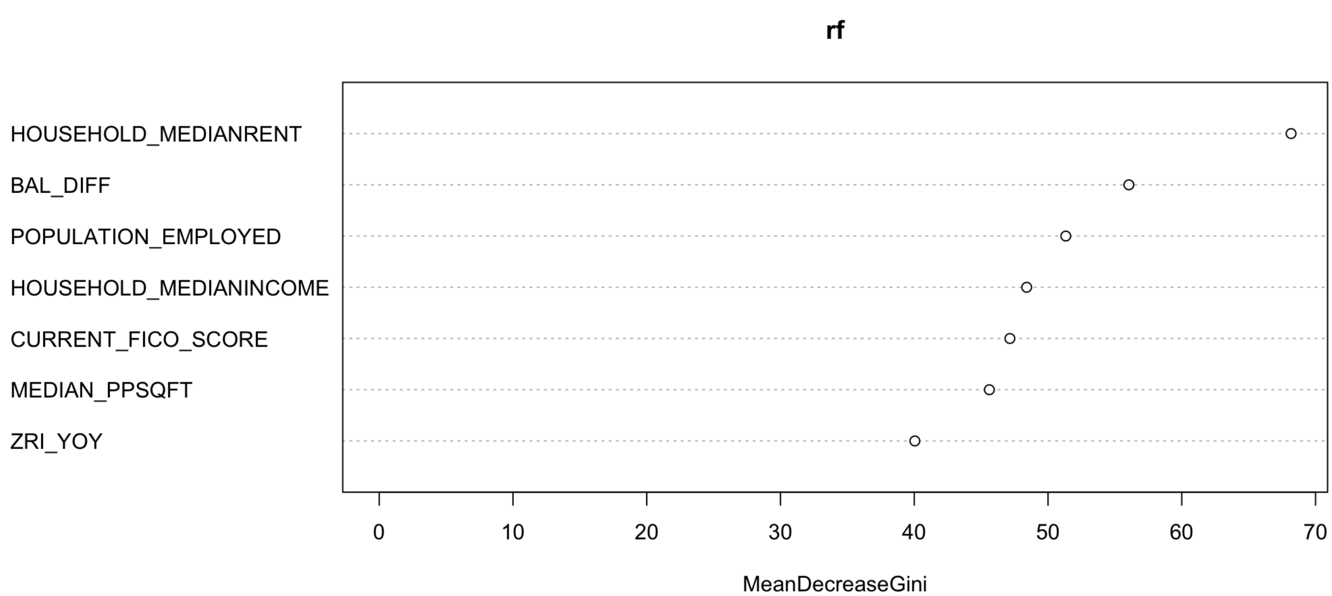
```
##train_data[, -c(5, ncol(train_data))] ## gets rid of these numbered columns starting,
 column i
rf = randomForest(y = train_data$Loan.Issued, x = train_data[, -3], ytest = test_data$Lo
an.Issued, xtest = test_data[, -3], ntree = 100, mtry = 3, keep.forest = TRUE, classwt =
 c(0.3,0.7))

rf
```

```
##
## Call:
##  randomForest(x = train_data[, -3], y = train_data$Loan.Issued,      xtest = test_dat
a[, -3], ytest = test_data$Loan.Issued, ntree = 100,      mtry = 3, classwt = c(0.3, 0.
7), keep.forest = TRUE)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 6.99%
## Confusion matrix:
##    0   1 class.error
## 0 42  34  0.44736842
## 1 28 783  0.03452528
##                 Test set error rate: 6.13%
## Confusion matrix:
##    0   1 class.error
## 0 20  18  0.47368421
## 1 10 409  0.02386635
```

Now, we see that the training error is 7.44% and the test error rate is 5.91%, an improvement from the test error rate of 6.35% before. More importantly, we have reduced the classification error of the non-loan issues from over 44.2% to 32.6%. This is really important because from the bank's perspective, you have to make sure that the loan you issue, for a commitment as large as housing, can be followed through.

Moreover, when we plot the variable importance plot

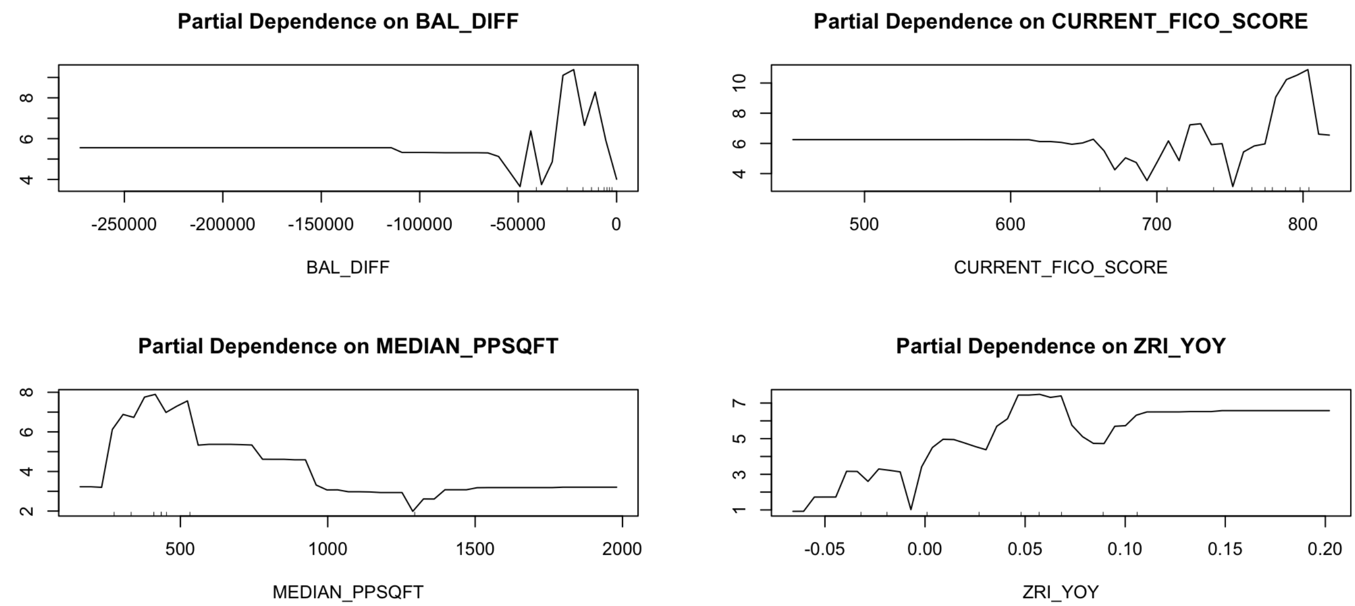
```
varImpPlot(rf,type=2)
```



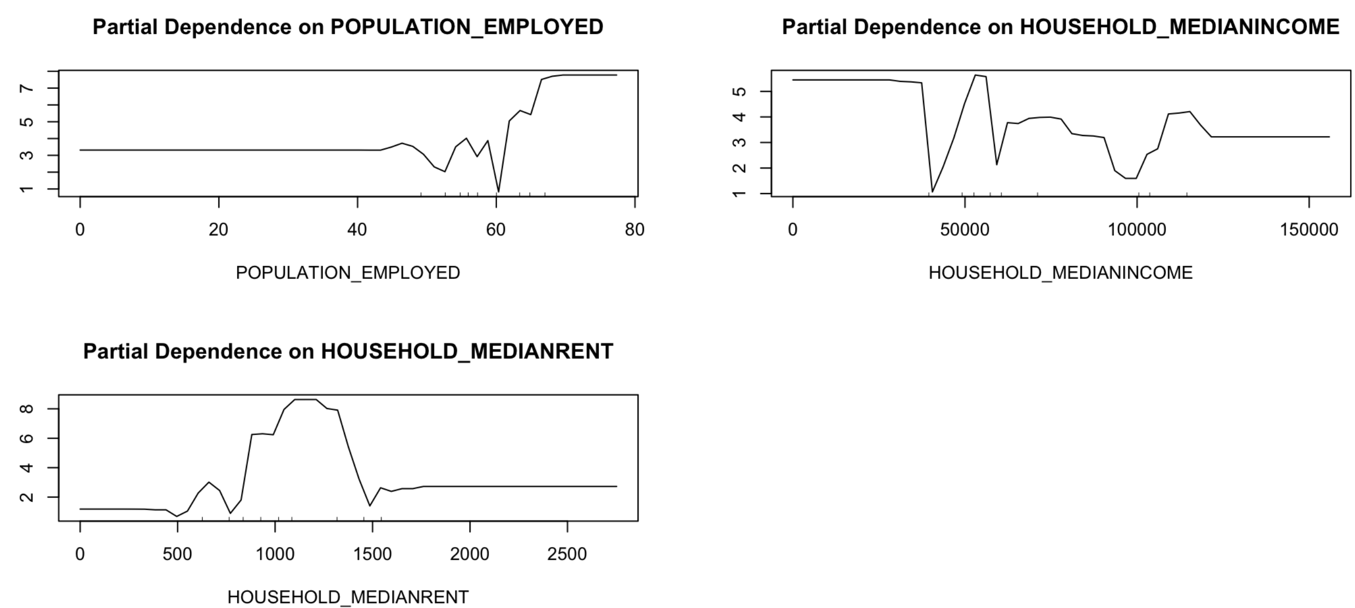
We see that now household median rent of the zip code is the most important factor as well as the population that is employed for that zip code. Furthermore, we see that the gap in the importance for the features is now smaller.

Now, let's check the partial dependence plots for the 7 variables:

```
par(mfrow=c(2,2))
partialPlot(rf, train_data, BAL_DIFF, 1)
partialPlot(rf, train_data, CURRENT_FICO_SCORE, 1)
partialPlot(rf, train_data, MEDIAN_PPSQFT, 1)
partialPlot(rf, train_data, ZRI_YOY, 1)
```




```
par(mfrow=c(2,2))
partialPlot(rf, train_data, POPULATION_EMPLOYED, 1)
partialPlot(rf, train_data, HOUSEHOLD_MEDIANINCOME, 1)
partialPlot(rf, train_data, HOUSEHOLD_MEDIANRENT, 1)
```



And lastly, we want to make a prediction on the row where the loan ID is 73622 and the house purchase is $500,000.

```
row <- data[data$LOANID == 73622,]
prob <- predict(rf,row,type="prob")
prob
```

```
##        0    1
## 701 0.96 0.04
## attr(,"class")
## [1] "matrix" "votes"
```

Conclusions

From our partial dependence plots, we don't really care about the actual y values in the partial dependence plots- we care more about their trends. We see that:

- People who were looking to get loans for properties in zipcodes with higher employment were more likely to get the loan issued. This could be because of the fact that in New York, a lot of high-paying jobs tend to be tedious/time-consuming. People will want to live near where they work in that case. Knowing that the person is hard working and is living in a job friendly area is a reason for higher loan issuing.
- For the zipcodes with high household median income and high household median rent, we see that the likelihood to get a loan issued is actually lower than if the median income/median rent was slightly more middle range (50,000-100,000 for income and 700-1500 for median rent). While this may be bizarre, we actually may conclude that there may be people wanting a housing loan for a lifestyle they simply cannot sustain or pay up. This might be lower/middle class people wanting to live an upper scale life by taking out loans. Nonetheless, banks wouldn't want to lend since that money might not come back. If I had more time, I would run another iteration of random forest modeling with the same model but with just one of household median income and household median rent. It seems that the two are somewhat correlated.
- Likelihood to give out loan increases as the current fico score increases, which is pretty intuitive
- Likelihood to give out loan decreases as the median price per square foot increases generally speaking. This may not seem intuitive, but it goes back to the fact that banks want to lend out money for purchases that they feel confident that they can get the money back for. Areas with higher price per square feet are expensive, and banks might be more reluctant to lend.
- One of the most interesting factors to analyze is the year over year for the ZRI, which is a house pricing index. We see that as the YOY grows from around -5% to 5%, the likelihood to give out a housing load grows. Yet, after the 5% YOY point, the likelihood drops. This could be due to the fact that rapidly growing areas in terms of demand may also carry higher risks (more volatile).

Lastly, we saw from our prediction of the request for a loan for a house price of $500,000 and loan id 73622 that our random forest model would decide that there is 0.93 probability that it is 0 and 0.07 probability that it is 1. Thus, we would conclude that we should not give out the loan for that purchase. Looking back through that dataset, we see that the person making the purchase has a recent drop in FICO score, has a low household income, and also is looking to make a housing purchase that is fairly pricey. From the story that we've told from our random forest model, these characteristics surely sound like reasons to not give out the loan.