Untitled

Kavana Manvi KrishnaMurthy

2024-04-30

Problem 1

For this problem, you will load and perform some cleaning steps on a dataset in the provided BankData.csv, which is data about loan approvals from a bank in Japan (it has been modified from the original for our purposes in class, so use the provided version). Specifically, you will use visualization to examine the variables and normalization, binning and smoothing to change them in particular ways.

a. Visualize the distributions of the variables in this data. You can choose bar graphs, histograms and density plots. Make appropriate choices given each type of variables and be careful when selecting parameters like the number of bins for the histograms. Note there are some numerical variables and some categorical ones. The ones labeled as a 'bool' are Boolean variables, meaning they are only true or false and are thus a special type of categorical. Checking all the distributions with visualization and summary statistics is a typical step when beginning to work with new data.

Load and look at the summary of BankData dataset.

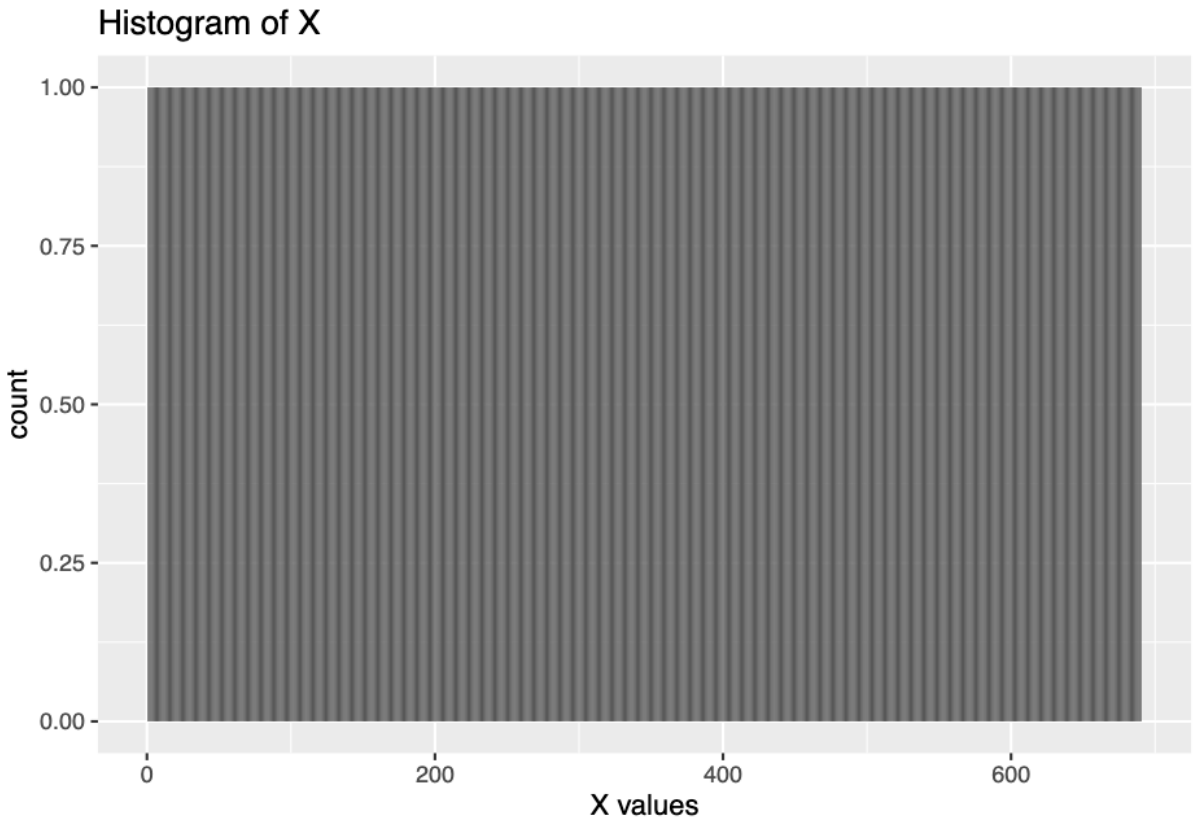
```
bank <-read.csv("/Users/kavanamanvi/Desktop/FDS/HW2/BankData.csv")
summary(bank)
```

```
##        X             cont1           cont2            cont3
##  Min.   :  1.0   Min.   :13.75   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:173.2   1st Qu.:22.60   1st Qu.: 1.000   1st Qu.: 0.165
##  Median :345.5   Median :28.46   Median : 2.750   Median : 1.000
##  Mean   :345.5   Mean   :31.57   Mean   : 4.759   Mean   : 2.223
##  3rd Qu.:517.8   3rd Qu.:38.23   3rd Qu.: 7.207   3rd Qu.: 2.625
##  Max.   :690.0   Max.   :80.25   Max.   :28.000   Max.   :28.500
##                  NA's   :12
##     bool1              bool2              cont4         bool3
##  Length:690         Length:690         Min.   : 0.0   Length:690
##  Class :character   Class :character   1st Qu.: 0.0   Class :character
##  Mode  :character   Mode  :character   Median : 0.0   Mode  :character
##                                        Mean   : 2.4
##                                        3rd Qu.: 3.0
##                                        Max.   :67.0
##
##      cont5          cont6            approval          credit.score
##  Min.   :   0   Min.   :     0.0   Length:690         Min.   :583.7
##  1st Qu.:  75   1st Qu.:     0.0   Class :character   1st Qu.:666.7
##  Median : 160   Median :     5.0   Mode  :character   Median :697.3
##  Mean   : 184   Mean   :  1017.4                      Mean   :696.4
```

```
##  3rd Qu.: 276   3rd Qu.:   395.5                    3rd Qu.:726.4
##  Max.   :2000   Max.   :100000.0                    Max.   :806.0
##  NA's   :13
##       ages
##  Min.   :11.00
##  1st Qu.:31.00
##  Median :38.00
##  Mean   :39.67
##  3rd Qu.:48.00
##  Max.   :84.00
##
```

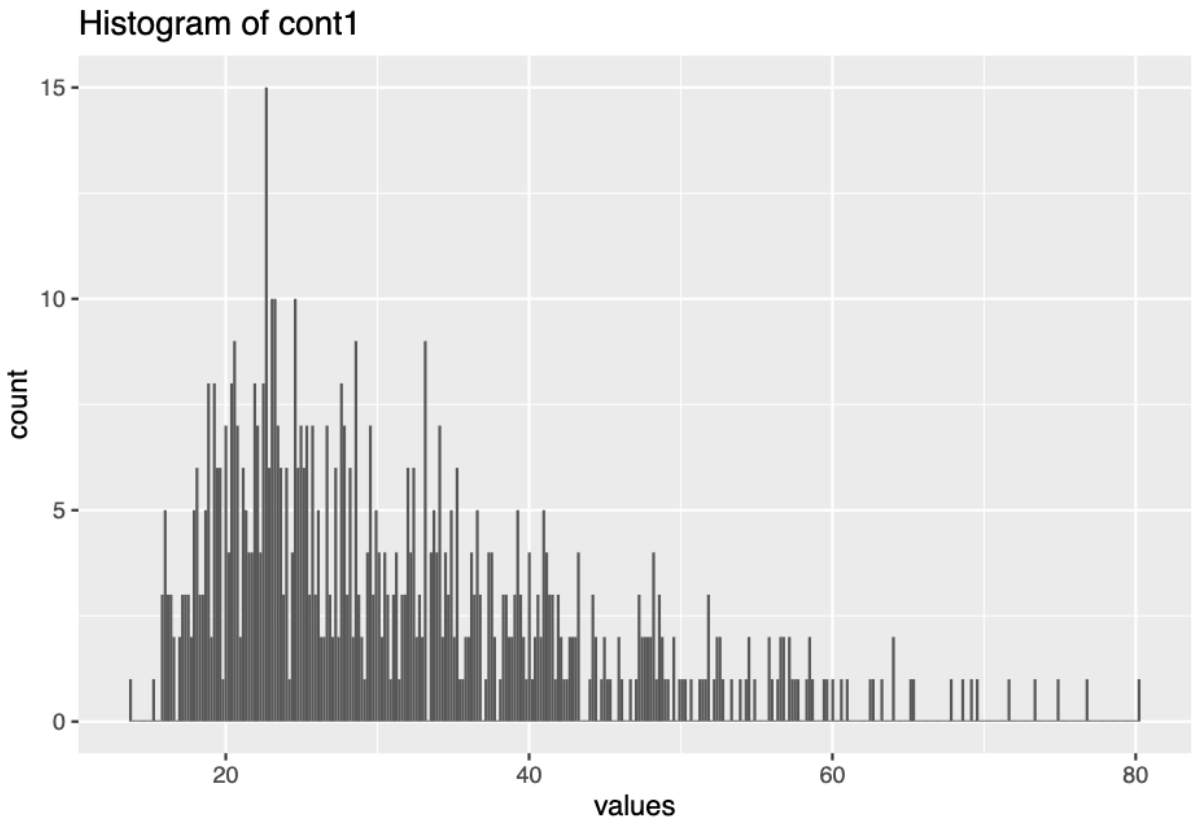
Histograms for numerical data

```
library("ggplot2")
ggplot(bank, aes(x = X)) +
geom_histogram(bins = length(unique(bank$X))) +
labs(title = "Histogram of X", x = "X values", y = "count")
```

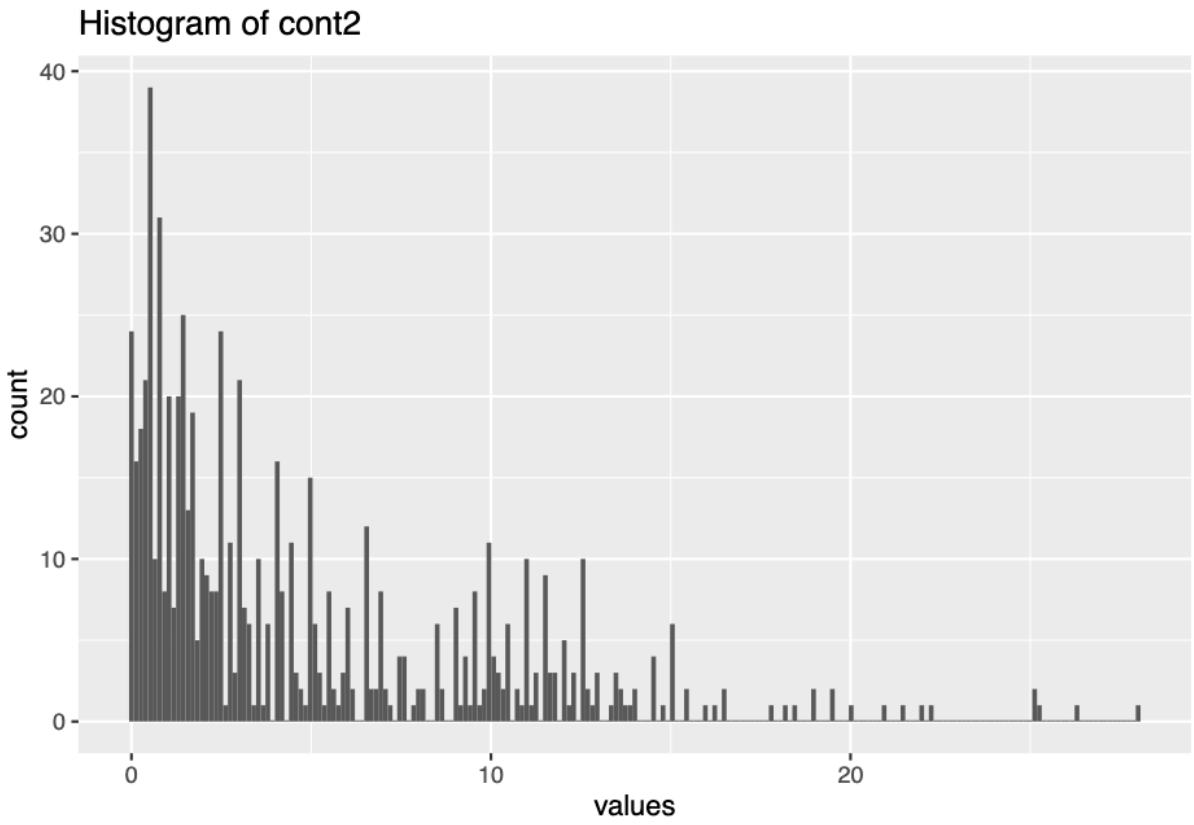



```
ggplot(bank, aes(x = cont1)) +
geom_histogram(bins = length(unique(bank$cont1))) +
labs(title = "Histogram of cont1", x = "values", y = "count")
```

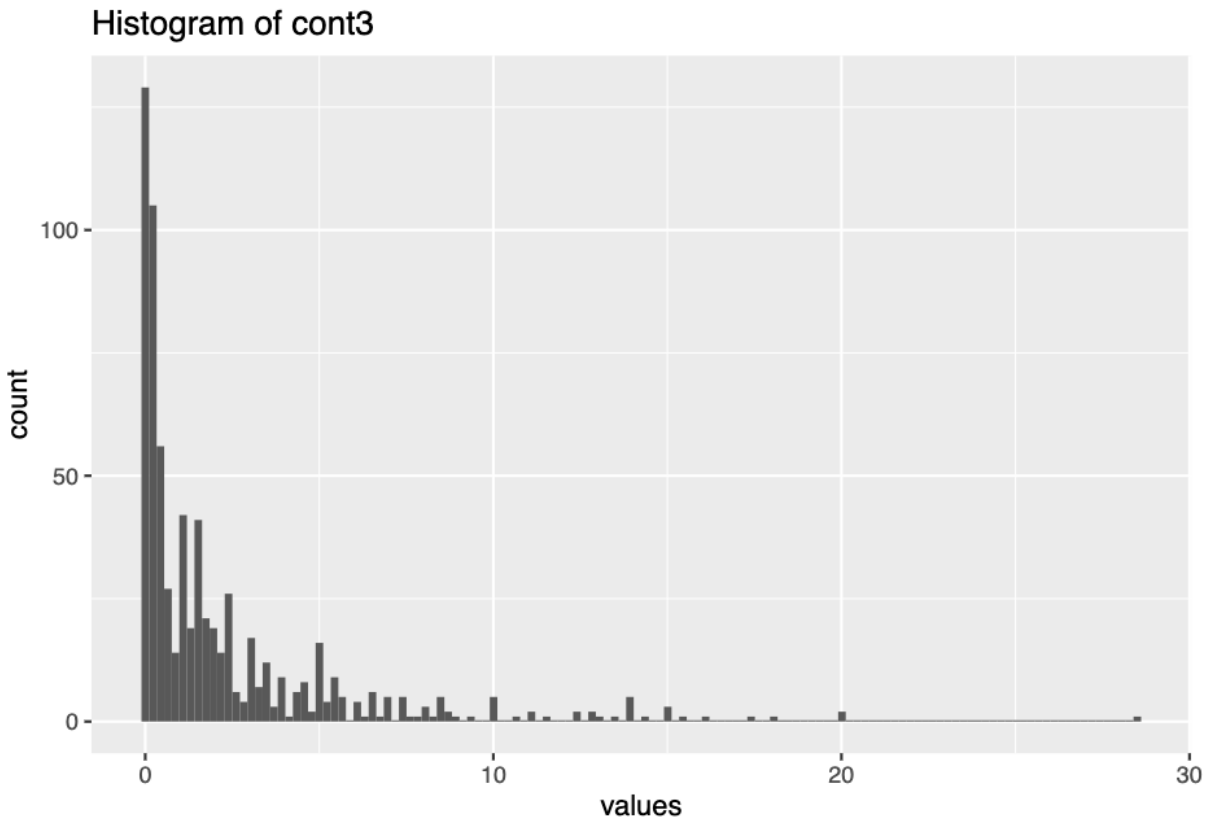
```
## Warning: Removed 12 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



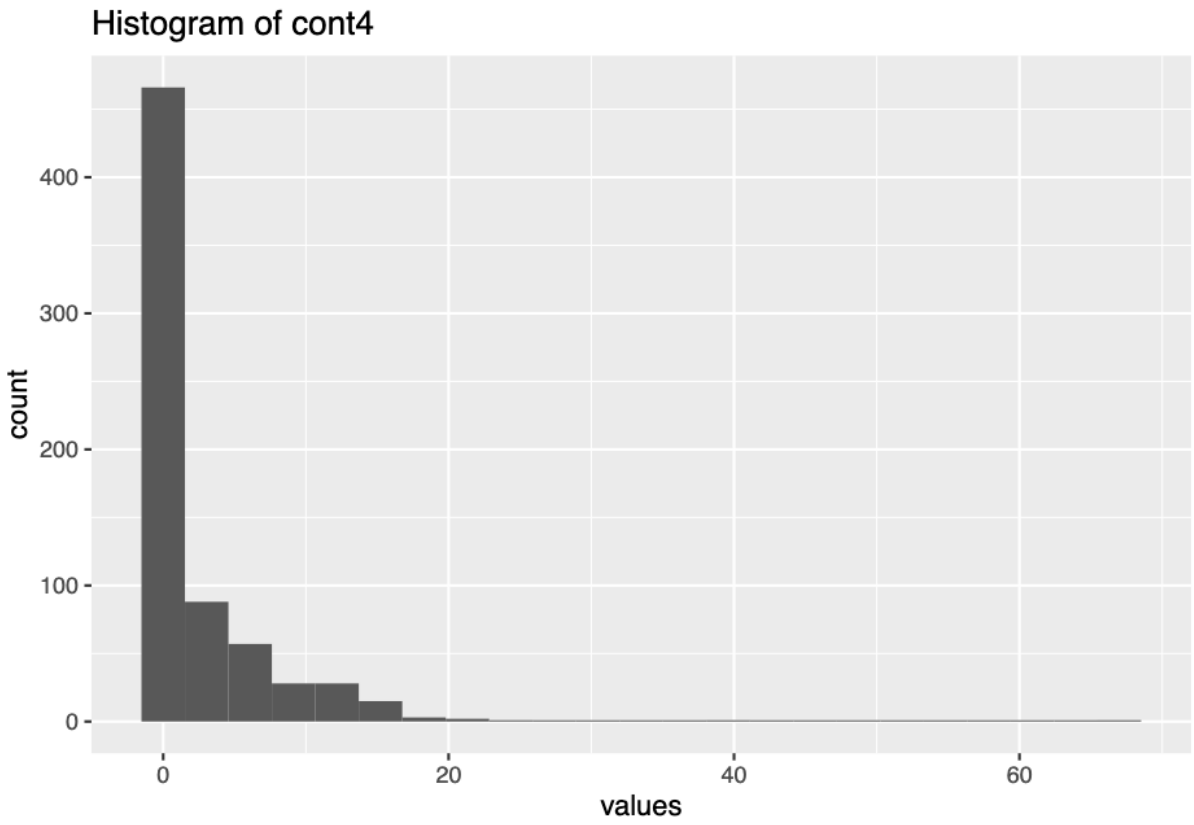

```
ggplot(bank, aes(x = cont2)) +
geom_histogram(bins = length(unique(bank$cont2))) +
labs(title = "Histogram of cont2", x = "values", y = "count")
```




```
ggplot(bank, aes(x = cont3)) +
geom_histogram(bins = length(unique(bank$cont3))) +
labs(title = "Histogram of cont3", x = "values", y = "count")
```



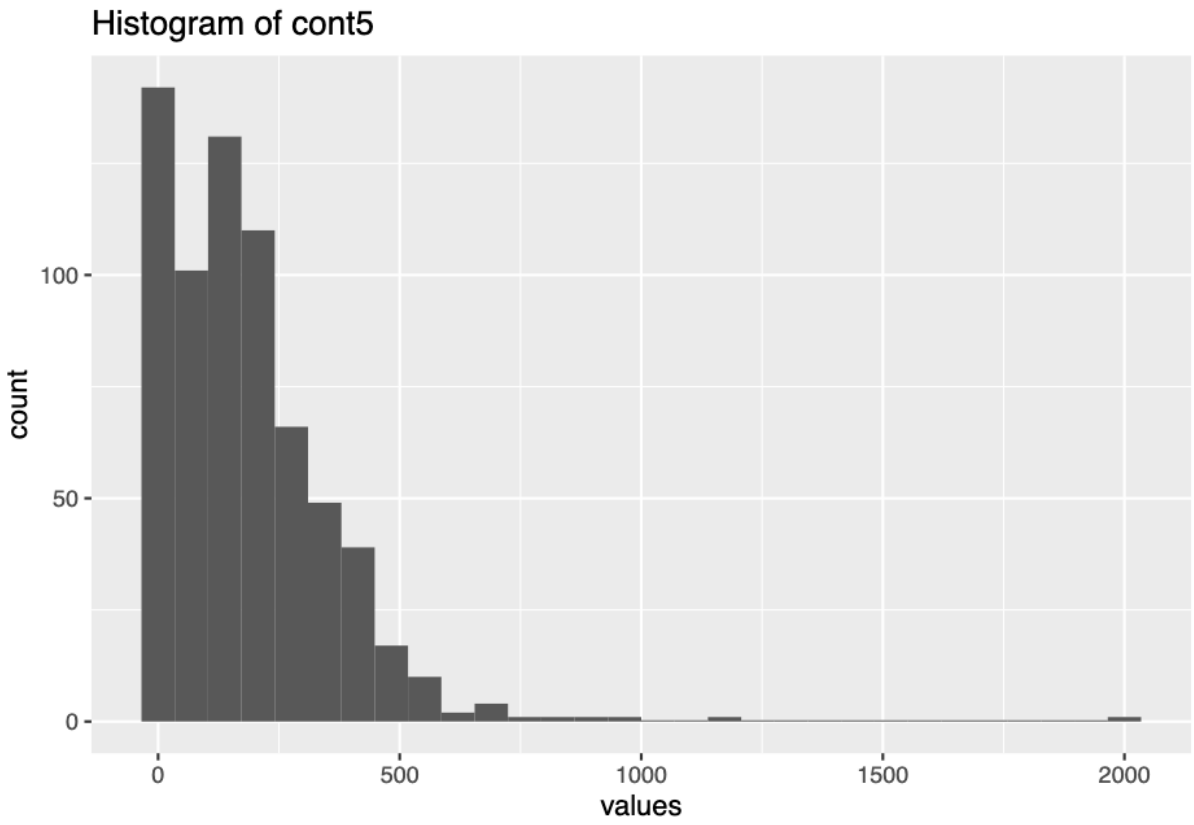

```
ggplot(bank, aes(x = cont4)) +
geom_histogram(bins = length(unique(bank$cont4))) +
labs(title = "Histogram of cont4", x = "values", y = "count")
```



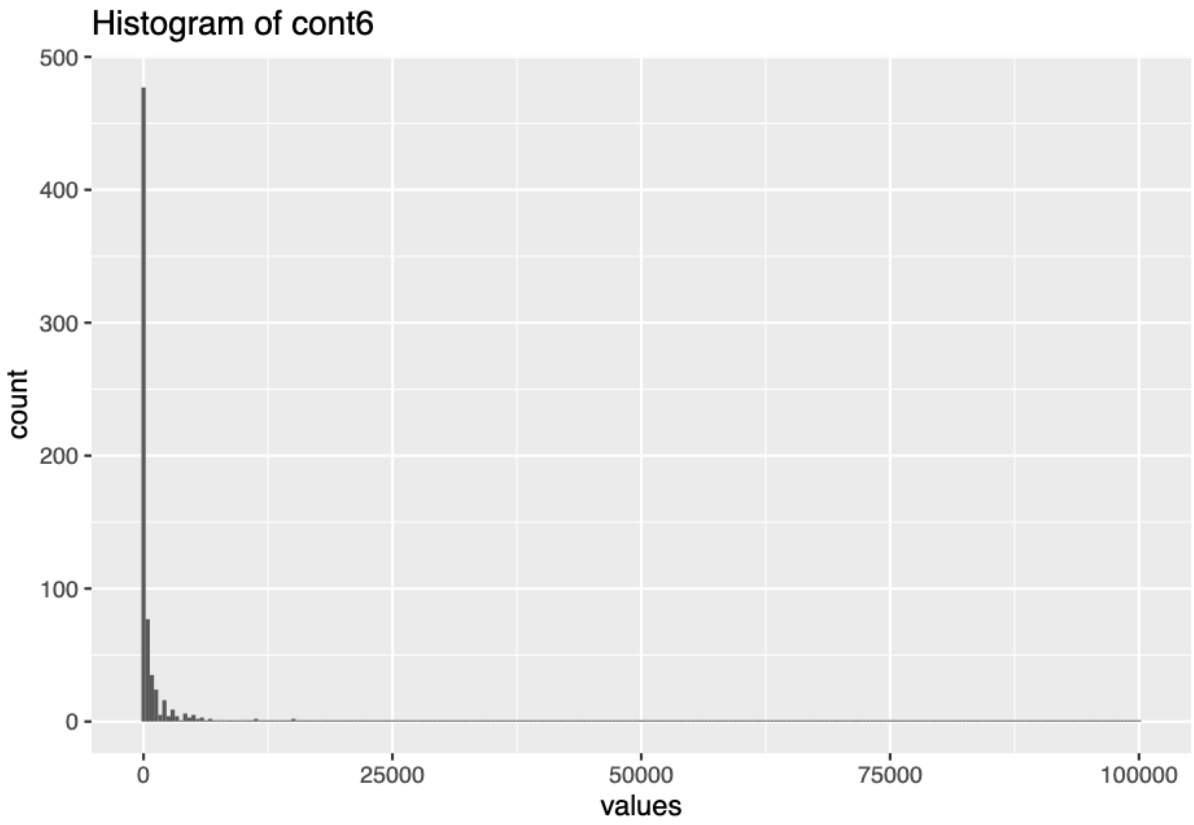

```
ggplot(bank, aes(x = cont5)) +
geom_histogram() +
labs(title = "Histogram of cont5", x = "values", y = "count")
```

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

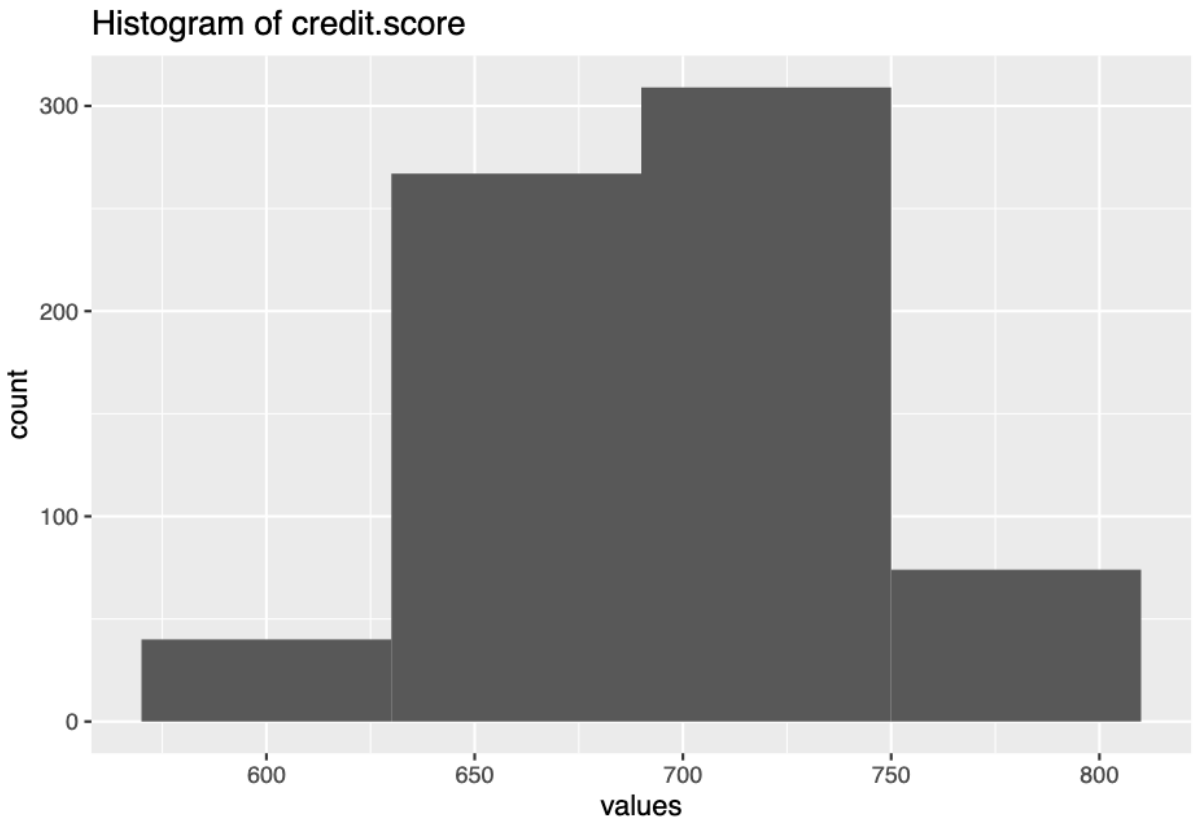
```
## Warning: Removed 13 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



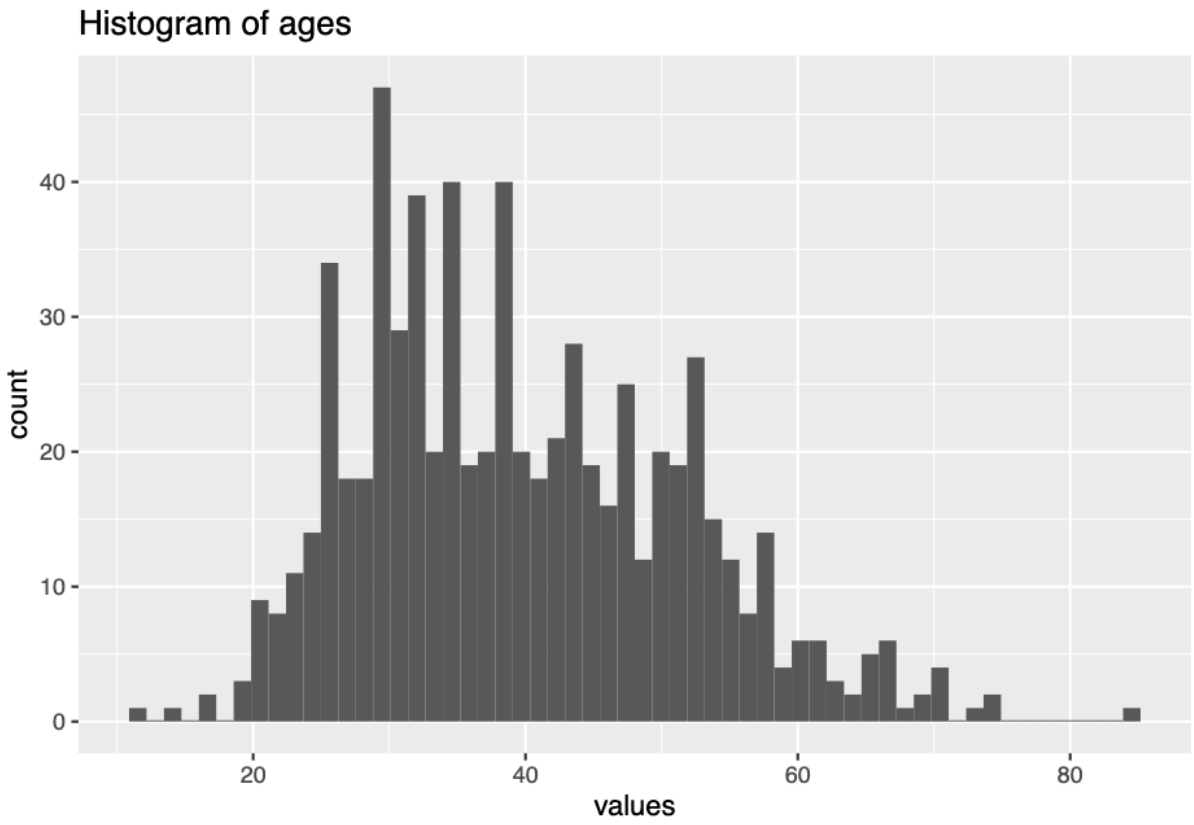

```
ggplot(bank, aes(x = cont6)) +
geom_histogram(bins = length(unique(bank$cont6))) +
labs(title = "Histogram of cont6", x = "values", y = "count")
```




```
ggplot(bank, aes(x = credit.score)) +
geom_histogram(binwidth = 60,bins = length(unique(bank$credit.score))) +
labs(title = "Histogram of credit.score", x = "values", y = "count")
```

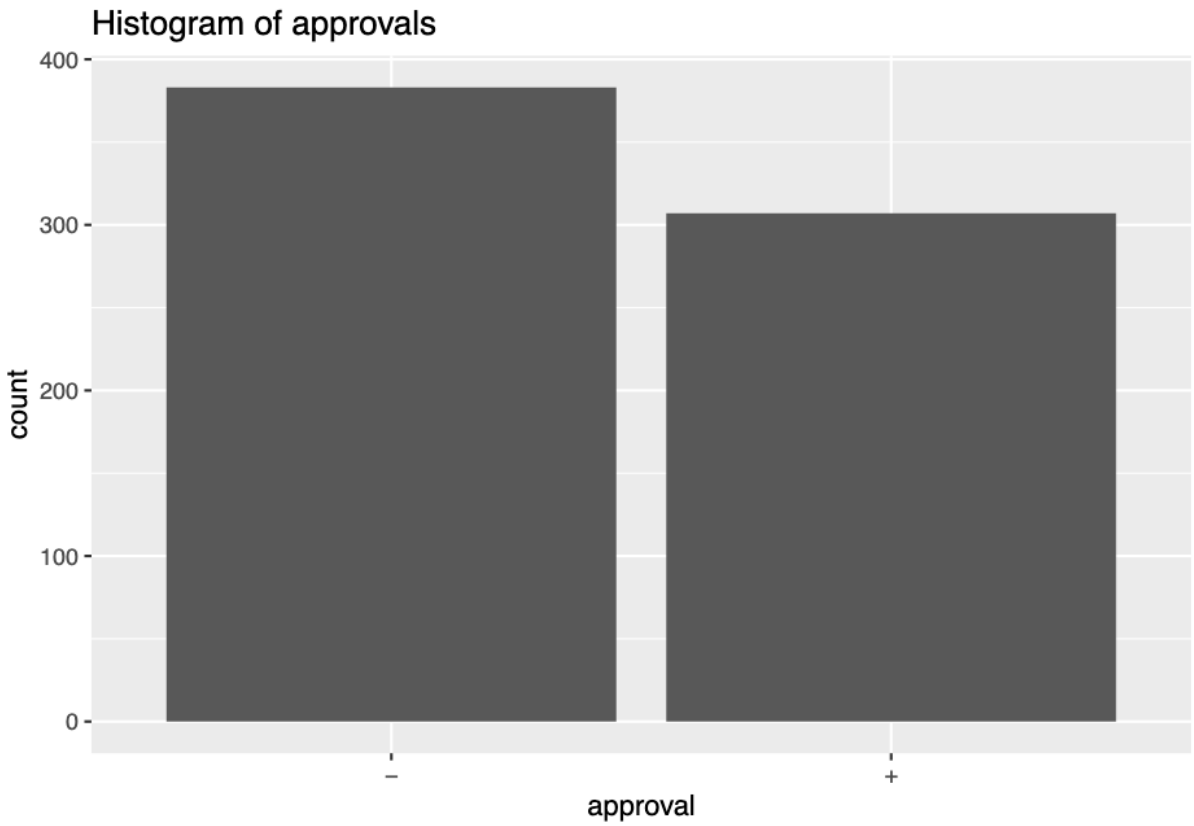



```
ggplot(bank, aes(x = ages)) +
geom_histogram(bins = length(unique(bank$ages))) +
labs(title = "Histogram of ages", x = "values", y = "count")
```

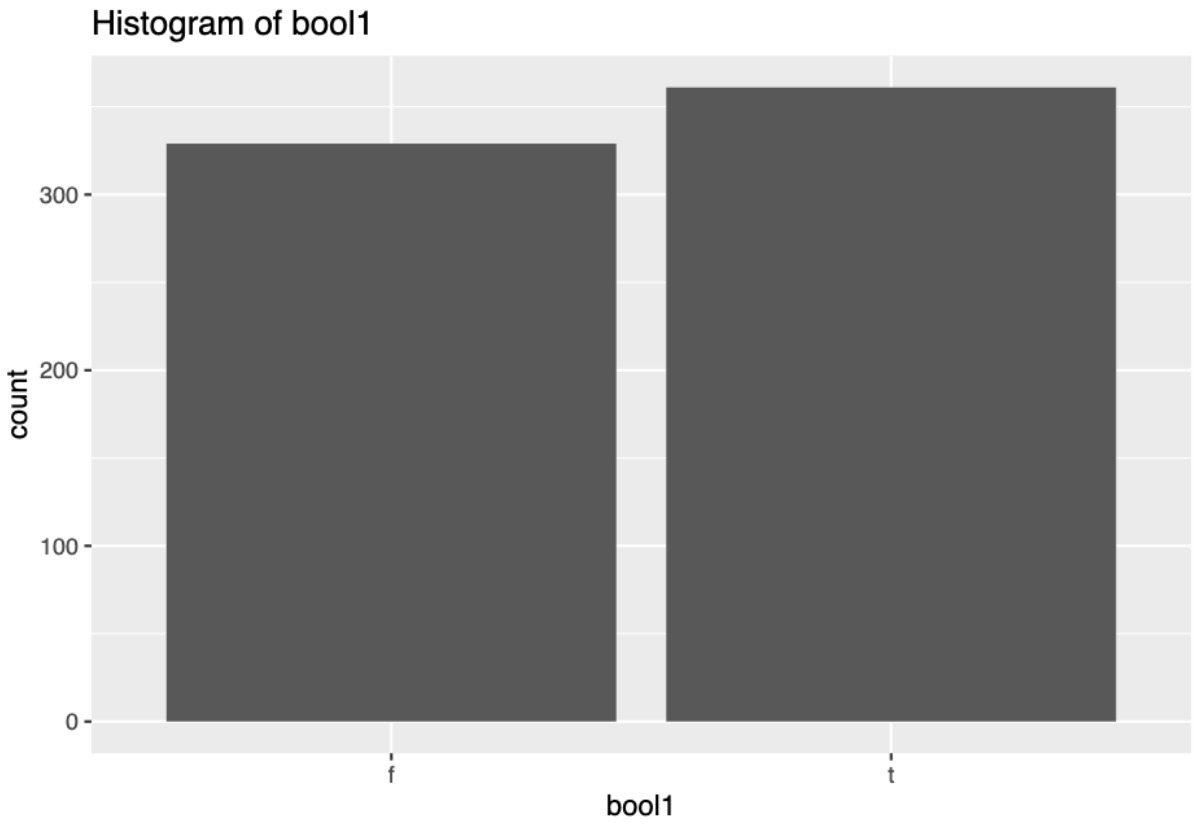


Bar graph for categorical variables

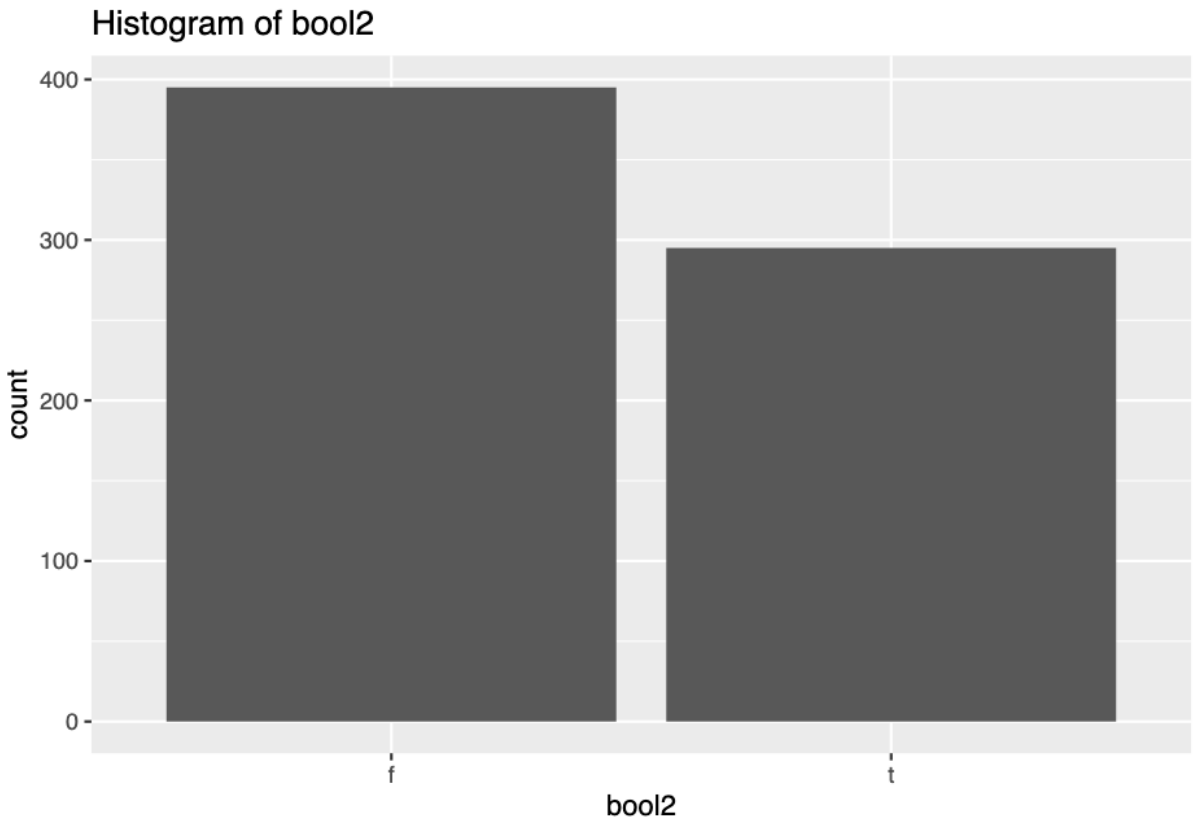
```
ggplot(bank, aes(x=approval)) + geom_bar()+
  labs(title = "Histogram of approvals")
```



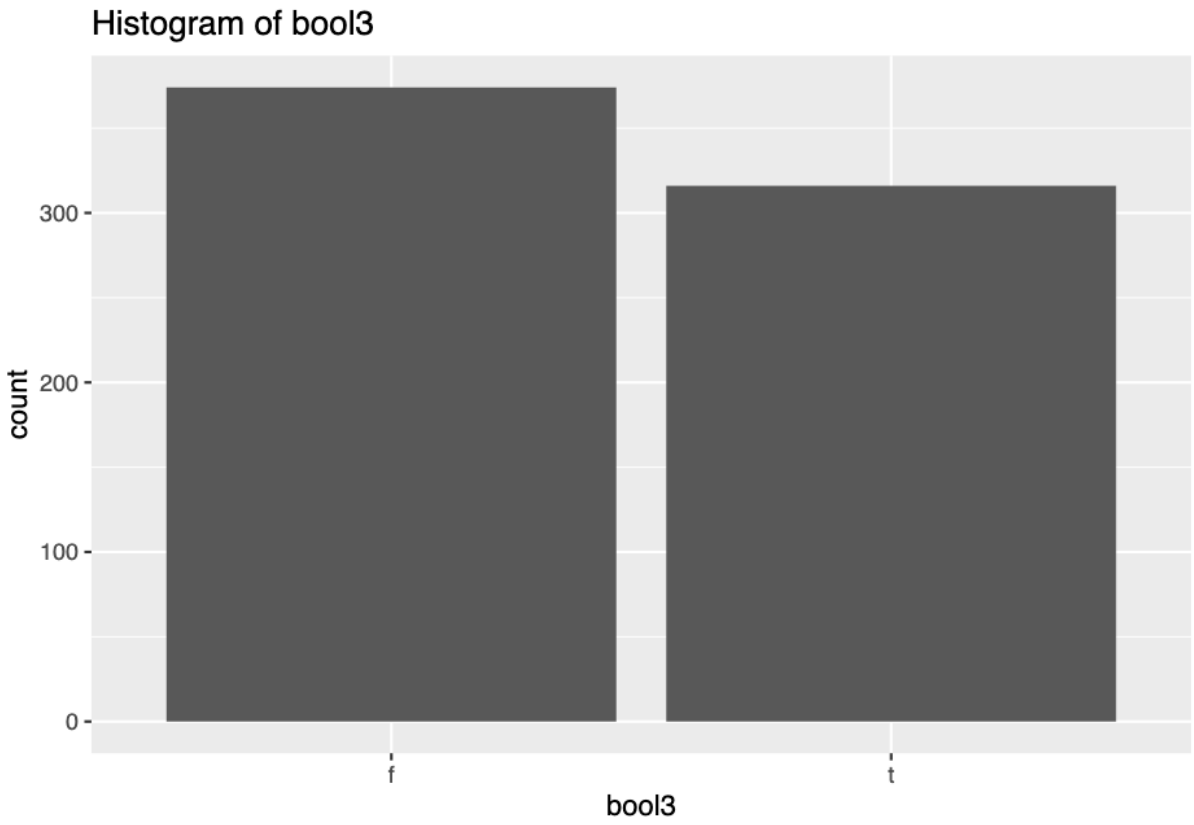

```
ggplot(bank, aes(x=bool1)) + geom_bar() +
labs(title = "Histogram of bool1")
```




```
ggplot(bank, aes(x=bool2)) + geom_bar()+
labs(title = "Histogram of bool2")
```




```
ggplot(bank, aes(x=bool3)) + geom_bar()+
  labs(title = "Histogram of bool3")
```



b. Now apply normalization to some of these numerical distributions. Specifically, choose to apply z-score to one, min-max to another, and decimal scaling to a third. Explain your choices of which normalization applies to which variable in terms of what the variable means, what distribution itstarts with, and how the normalization will affect it.

Apply Z-score normalization to Age :For the Age feature, I would apply Z-score normalization. Z-score normalization is suitable for this variable because it does not contain extreme outliers, and it allows us to standardize the feature distribution to have a mean of 0 and a standard deviation of 1. This technique is particularly useful when dealing with algorithms that assume normally distributed data, such as many linear models. Unlike the min-max scaling technique, which restricts feature values to a specific range, Z-score normalization represents features in terms of the number of standard deviations they lie away from the mean.It reduces the range from 0 to 85 upto a range of -2 to 4.This technique is particularly useful when dealing with algorithms that assume normally distributed data, such as many linear models.

```
library(caret)
```

```
## Loading required package: lattice
```

```
# Convert bank$age to a data frame
age_df <- data.frame(age = bank$age)
# Preprocess the age variable
preproc_age <- preProcess(age_df, method = c("center", "scale"))
# Standardize the age variable
normalized_age <- predict(preproc_age, age_df)
```

```r
# Summary of the normalized age variable
summary(normalized_age)
```

```
##       age
##  Min.   :-2.4434
##  1st Qu.:-0.7391
##  Median :-0.1426
##  Mean   : 0.0000
##  3rd Qu.: 0.7095
##  Max.   : 3.7772
```

```r
bank$ages_zscore <- normalized_age$age
```

Apply decimal scaling to credit score:Decimal scaling normalization is used for the credit score feature in a bank dataset because it allows us to scale the feature values such that the largest absolute value becomes less than 1. In the context of a credit score, which typically ranges from a few hundred to a few thousand, decimal scaling ensures that the scale of the credit score is consistent across different datasets or features. This is important because the absolute magnitude of the credit score matters more than its specific scale.

Normalization affects the credit score by scaling it to a range where the largest absolute value is less than 1. This makes it easier to compare credit scores across different individuals or datasets, as the scale is consistent. Additionally, normalization can help improve the performance of machine learning algorithms that are sensitive to the scale of the features, such as linear models

```r
bank$credit.score_decimal <- bank$credit.score / 1000
```

Apply min-max scaling to cont6:Here the data is scaled to a fixed range (usually 0 to 1). The impact is that we end up with smaller standard deviations, which can suppress the effect of outliers. The code is very similar to standardization. Since it has a lot of outliers, min-max scaling supresses it.

```r
# Convert bank$cont6 to a data frame
cont6_df <- data.frame(cont6 = bank$cont6)

# Preprocess the cont6 variable
preproc_cont6 <- preProcess(cont6_df, method = c("range"))

# Apply the min-max scaling to cont6
normalized_cont6 <- predict(preproc_cont6, cont6_df)

# Summary of the normalized cont6 variable
summary(normalized_cont6)
```

```
##      cont6
##  Min.   :0.000000
##  1st Qu.:0.000000
##  Median :0.000050
##  Mean   :0.010174
##  3rd Qu.:0.003955
##  Max.   :1.000000
```

```
bank$cont6_minmax <-normalized_cont6$cont6
```

Print the first 6 rows of normalized data

```
head(bank[, c("ages", "ages_zscore", "credit.score", "credit.score_decimal", "cont6", "cont6_minmax")])
```

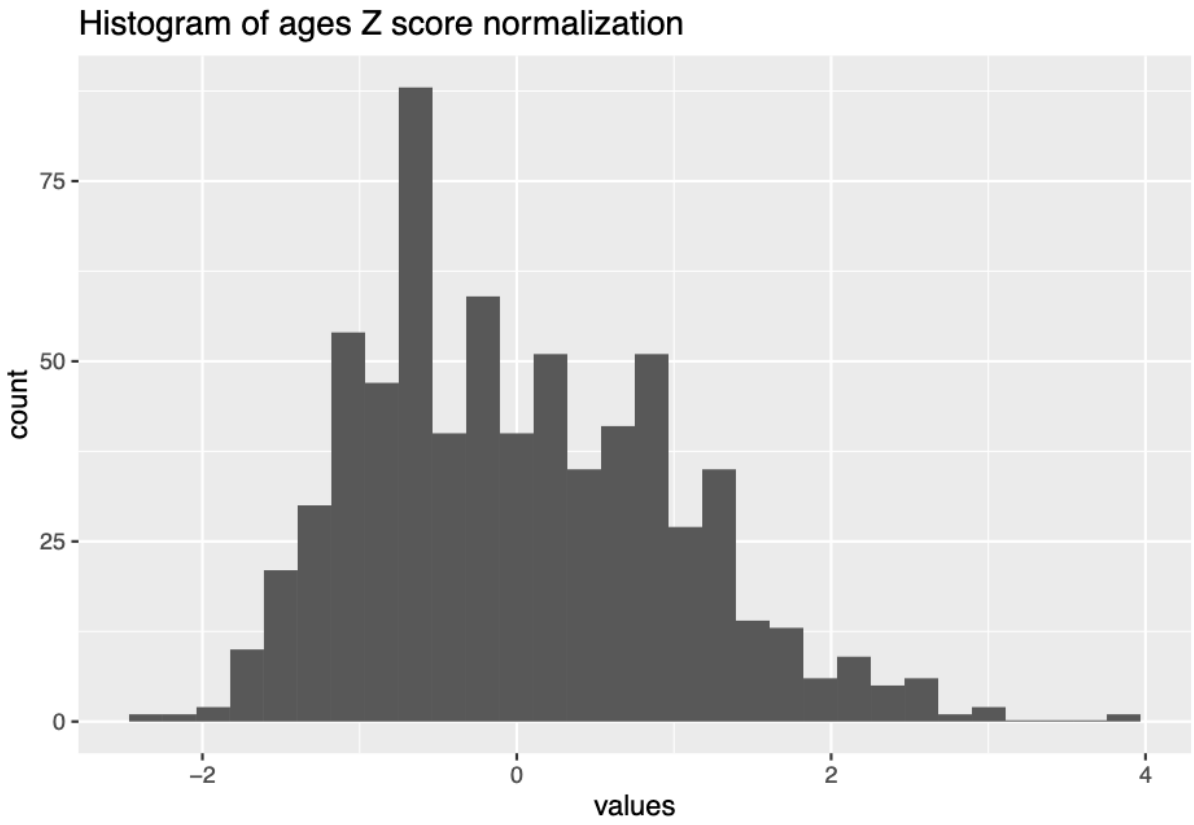
```
##   ages ages_zscore credit.score credit.score_decimal cont6 cont6_minmax
## 1   42   0.1982145       664.60              0.66460     0      0.00000
## 2   54   1.2207788       693.88              0.69388   560      0.00560
## 3   29  -0.9095635       621.82              0.62182   824      0.00824
## 4   58   1.5616335       653.97              0.65397     3      0.00003
## 5   65   2.1581294       670.26              0.67026     0      0.00000
## 6   61   1.8172746       672.16              0.67216     0      0.00000
```

c. Visualize the new distributions for the variables that have been normalized. What has changed from the previous visualization?

Apply Z-score normalization to Age:Z-score normalization represents features in terms of the number of standard deviations they lie away from the mean.It reduces the range from 0 to 85 upto a range of -2 to 4.This technique is particularly useful when dealing with algorithms that assume normally distributed data, such as many linear models.

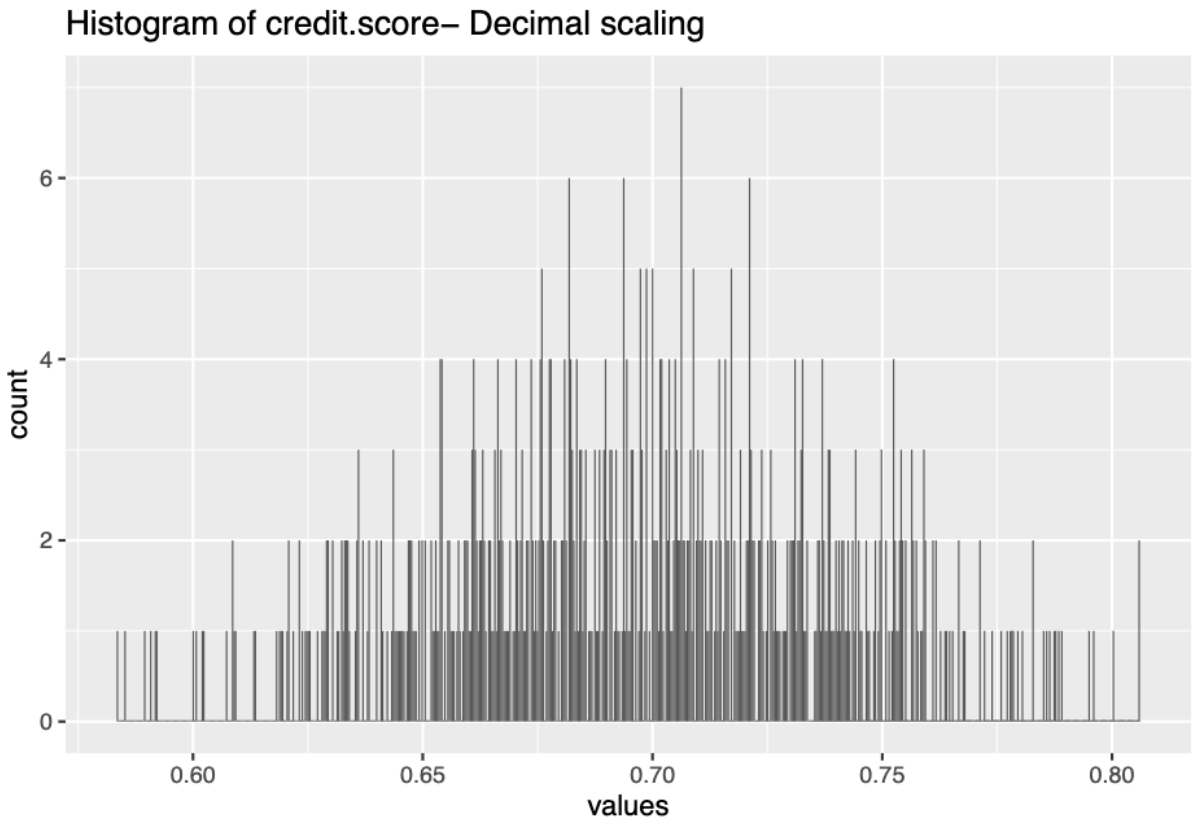
```
ggplot(bank, aes(x = ages_zscore)) +
geom_histogram() +
labs(title = "Histogram of ages Z score normalization", x = "values", y = "count")
```

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



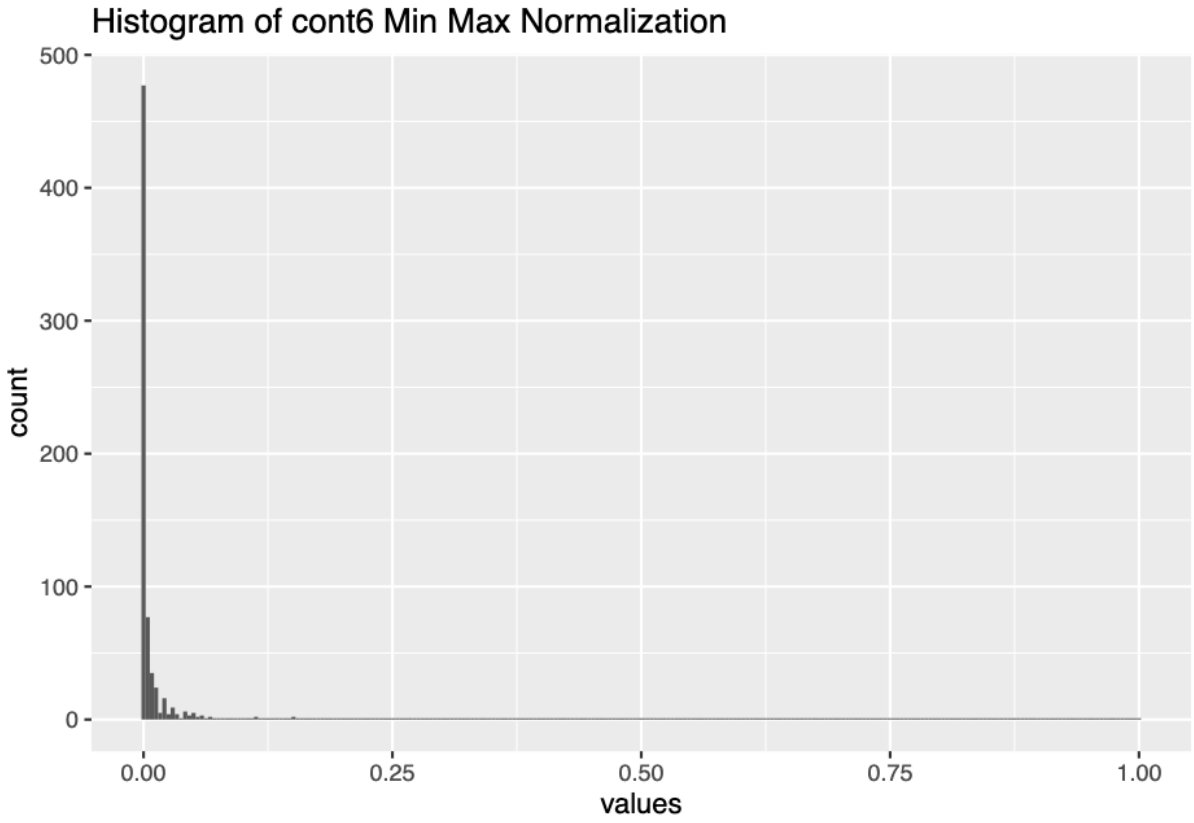
Apply decimal scaling to credit score:Decimal scaling normalization is used for the credit score feature in a bank dataset because it allows us to scale the feature values such that the largest absolute value becomes less than 1.Normalization affects the credit score by scaling it to a range where the largest absolute value is less than 1.

```
ggplot(bank, aes(x = credit.score_decimal)) +
geom_histogram(bins = length(unique(bank$credit.score_decimal))) +
labs(title = "Histogram of credit.score- Decimal scaling", x = "values", y = "count")
```



Apply min-max scaling to cont2:It brings the values to a range of -1 to 1.in-max scaling to a range involves converting floating-point feature values from their original range (-30 to 30) into a standardized range, typically between 0 and 1.

```
ggplot(bank, aes(x = cont6_minmax)) +
geom_histogram(bins = length(unique(bank$cont6_minmax))) +
labs(title = "Histogram of cont6 Min Max Normalization", x = "values", y = "count")
```



d. Choose one of the numerical variables to work with for this problem. Let's call it v. Create a new variable called v_bins that is a binned version of that variable. This v_bins will have a new set of values like low, medium, high. Choose the actual new values (you don't need to use low, medium, high) and the ranges of v that they represent based on your understanding of v from your visualizations. You can use equal depth, equal width or custom ranges. Explain your choices: why did you choose to create that number of values and those particular ranges?

```
summary(bank$credit.score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   583.7   666.7   697.3   696.4   726.4   806.0
```

```
ggplot(bank, aes(y = credit.score)) +
  geom_boxplot() +
  labs(title = "Box Plot of Credit Score", y = "Credit Score")
```

Given that credit.score ranges from 583.7 to 806.0, we can divide it into three bins: low, medium, and high. The ranges for these bins can be defined as follows:

Low: Below the 1st quartile (Q1) value of credit.score (i.e., less than 666.7) Medium: Between the 1st quartile (Q1) and the 3rd quartile (Q3) values of credit.score (i.e., between 666.7 and 726.4) High: Above the 3rd quartile (Q3) value of credit.score (i.e., greater than 726.4) We choose these ranges to evenly distribute the data into three categories based on quartiles, which gives us a good representation of the data distribution. Additionally, using quartiles ensures that each bin contains a similar number of observations, which can help in creating meaningful categories for analysis.

```
library(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
```

```
bank <- bank %>%
  mutate(credit.score_bins = cut(credit.score,
```

```
                                  breaks=c(-Inf, 666.7, 726.4, Inf),
                                  labels=c("low", "medium", "high")))
head(bank$credit.score_bins)
```

```
## [1] low    medium low    low    medium medium
## Levels: low medium high
```

e. Building on (d), use v_bins to create a smoothed version of v. Choose a smoothing strategy to create a numerical version of the binned variable and explain your choices.

Create Bins: Use the cut function to create equal width bins for the credit.score variable and store the result in a new column credit.score_bins. Calculate Mean for Each Bin: Calculate the mean value of credit.score for each bin (low, medium, high) and store the result in separate data frames for each bin. Replace Original Values: Replace the original credit.score values with the mean value of each bin for the corresponding credit.score_bins. Combine Results: Combine the separate data frames for each bin into a single data frame to get the smoothed version of credit.score_bins.

```
# Create bins for credit score
bank <- bank %>%
  mutate(credit.score_bins = cut(credit.score, breaks = 3,
                                 labels = c("low", "medium", "high")))

# Calculate mean value for each bin and replace the original
#credit score with the mean
low <- bank %>%
  filter(credit.score_bins == 'low') %>%
  mutate(credit.score = mean(credit.score, na.rm = TRUE))

medium <- bank %>%
  filter(credit.score_bins == 'medium') %>%
  mutate(credit.score = mean(credit.score, na.rm = TRUE))

high <- bank %>%
  filter(credit.score_bins == 'high') %>%
  mutate(credit.score = mean(credit.score, na.rm = TRUE))

# Combine the results
smoothed_bank <- bind_rows(list(low, medium, high))
bind_rows(list(low, medium, high))
```

```
##       X cont1  cont2  cont3 bool1 bool2 cont4 bool3 cont5  cont6 approval
## 1     3 24.50  0.500  1.500     t     f     0     f   280   824        +
## 2     4 27.83  1.540  3.750     t     t     5     t   100     3        +
## 3    16 36.67  4.415  0.250     t     t    10     t   320     0        +
## 4    19 21.83  0.250  0.665     t     f     0     t     0     0        +
## 5    26 15.83  0.585  1.500     t     t     2     f   100     0        +
## 6    28 56.58 18.500 15.000     t     t    17     t     0     0        +
## 7    33 49.50  7.585  7.585     t     t    15     t     0  5000        +
## 8    40 54.58  9.415 14.415     t     t    11     t    30   300        +
## 9    41 34.17  9.170  4.500     t     t    12     t     0   221        +
## 10   46 54.33  6.750  2.625     t     t    11     t     0   284        +
## 11   47 41.00  2.040  0.125     t     t    23     t   455  1236        +
```

```
## 12   56 23.33 11.625  0.835    t    f    0    t  160    300       +
## 13   59 35.17  4.500  5.750    f    f    0    t  711      0       +
## 14   63 23.42  0.790  1.500    t    t    2    t   80    400       +
## 15   65 26.67  4.250  4.290    t    t    1    t  120      0       +
## 16   69 19.42  6.500  1.460    t    t    7    f   80   2954       +
## 17   77 34.08  6.500  0.125    t    f    0    t  443      0       -
## 18   80 21.50  9.750  0.250    t    f    0    f  140      0       -
## 19   99 22.50 11.000  3.000    t    f    0    t  268      0       -
## 20  104 25.00 12.000  2.250    t    t    2    t  120      5       -
## 21  110 19.75  0.750  0.795    t    t    5    t  140      5       -
## 22  121 39.92  6.210  0.040    t    t    1    f  200    300       +
## 23  129 34.42  4.250  3.250    t    t    2    f  274    610       +
## 24  130 28.42  3.500  0.835    t    f    0    f  280      0       +
## 25  138 33.58  2.750  4.250    t    t    6    f  204      0       +
## 26  146 32.83  2.500  2.750    t    t    6    f  160   2072       +
## 27  154 23.08  2.500  1.085    t    t   11    t   60   2184       +
## 28  162 44.00  2.000  1.750    t    t    2    t    0     15       +
## 29  188 40.58  5.000  5.000    t    t    7    f    0   3065       +
## 30  189 28.67  1.040  2.500    t    t    5    t  300   1430       +
## 31  199 27.58  2.040  2.000    t    t    3    t  370    560       +
## 32  207 71.58  0.000  0.000    f    f    0    f   NA      0       +
## 33  209 35.17  2.500  4.500    t    t    7    f  150   1270       +
## 34  212 24.33  6.625  5.500    t    f    0    t  100      0       +
## 35  239 42.83  4.625  4.580    t    f    0    f    0      0       +
## 36  254 23.17 11.125  0.460    t    t    1    f  100      0       +
## 37  263 48.17  1.335  0.335    f    f    0    f    0    120       -
## 38  265 50.75  0.585  0.000    f    f    0    f  145      0       -
## 39  275 30.67  2.500  2.250    f    f    0    t  340      0       -
## 40  276 18.58  5.710  0.540    f    f    0    f  120      0       -
## 41  277 19.17  5.415  0.290    f    f    0    f   80    484       -
## 42  286 17.58 10.000  0.165    f    t    1    f  120      1       -
## 43  288 29.50  0.580  0.290    f    t    1    f  340   2803       -
## 44  294 35.75  2.415  0.125    f    t    2    f  220      1       -
## 45  300 22.17 12.125  3.335    f    t    2    t  180    173       -
## 46  304 15.92  2.875  0.085    f    f    0    f  120      0       -
## 47  311 24.83  4.500  1.000    f    f    0    t  360      6       -
## 48  312 19.00  1.750  2.335    f    f    0    t  112      6       -
## 49  318 17.50 22.000  0.000    f    f    0    t  450 100000       +
## 50  325 33.67  1.250  1.165    f    f    0    f  120      0       -
## 51  329 34.83  2.500  3.000    f    f    0    f  200      0       -
## 52  330    NA  4.000  0.085    f    f    0    t  411      0       -
## 53  331 20.42  0.000  0.000    f    f    0    f   NA      0       -
## 54  332 33.25  2.500  2.500    f    f    0    t    0      2       -
## 55  334 25.25 12.500  1.000    f    f    0    t  180   1062       -
## 56  335 34.75  2.500  0.500    f    f    0    f  348      0       -
## 57  336 27.67  0.750  0.165    f    f    0    t  220    251       -
## 58  348 26.75  4.500  2.500    f    f    0    f  200   1210       -
## 59  349 63.33  0.540  0.585    t    t    3    t  180      0       -
## 60  351 26.17  2.000  0.000    f    f    0    t  276      1       -
## 61  354 30.75  1.585  0.585    f    f    0    t    0      0       -
## 62  359 32.42  3.000  0.165    f    f    0    t  120      0       -
## 63  364 16.92  0.335  0.290    f    f    0    f  200      0       -
## 64  372 26.33 13.000  0.000    f    f    0    t  140   1110       -
## 65  383 24.33  2.500  4.500    f    f    0    f  200    456       -
```

```
## 66  388 21.17  0.000  0.500     f     f     0     t     0      0        -
## 67  396 33.08  1.625  0.540     f     f     0     t     0      0        -
## 68  398 23.58  0.585  0.125     f     f     0     f   120     87        -
## 69  403 51.92  6.500  3.085     f     f     0     t    73      0        -
## 70  418 23.58  1.790  0.540     f     f     0     t   136      1        -
## 71  422 20.42  1.085  1.500     f     f     0     f   108      7        -
## 72  423 29.42  1.250  1.750     f     f     0     f   200      0        -
## 73  424 26.17  0.835  1.165     f     f     0     f   100      0        -
## 74  426 24.58  1.250  0.250     f     f     0     f   110      0        -
## 75  431 51.83  3.000  1.500     f     f     0     f   180      4        -
## 76  443 30.58  2.710  0.125     f     f     0     t    80      0        -
## 77  444 17.25  3.000  0.040     f     f     0     t   160     40        -
## 78  461 24.50  2.415  0.000     f     f     0     f   120      0        -
## 79  469 22.08  2.335  0.750     f     f     0     f   180      0        -
## 80  471 21.92 11.665  0.085     f     f     0     f   320      5        -
## 81  472 21.08  4.125  0.040     f     f     0     f   140    100        -
## 82  478 39.17  2.500 10.000     f     f     0     t   200      0        -
## 83  491 47.25  0.750  2.750     t     t     1     f   333    892        +
## 84  499 25.75  0.500  1.460     t     t     5     t   312      0        +
## 85  511 13.75  4.000  1.750     t     t     2     t   120   1000        +
## 86  512 46.00  4.000  0.000     t     f     0     f   100    960        +
## 87  520 39.17  1.710  0.125     t     t     5     t   480      0        +
## 88  523 22.83  3.000  1.290     t     t     1     f   260    800        +
## 89  531 25.33  2.085  2.750     t     f     0     t   360      1        -
## 90  535 31.83  2.500  7.500     t     f     0     t   523      0        -
## 91  538 35.25  3.165  3.750     t     f     0     t   680      0        -
## 92  542 42.75  3.000  1.000     t     f     0     f     0    200        -
## 93  543 19.67 10.000  0.835     t     f     0     t   140      0        -
## 94  551 76.75 22.290 12.750     t     t     1     t     0    109        +
## 95  553 34.75 15.000  5.375     t     t     9     t     0    134        +
## 96  562 26.75  1.125  1.250     t     f     0     f     0   5298        +
## 97  566 20.83  3.000  0.040     t     f     0     f   100      0        +
## 98  567 23.08 11.500  2.125     t     t    11     t   290    284        +
## 99  575 20.33 10.000  1.000     t     t     4     f    50   1465        +
## 100 579 39.17  1.625  1.500     t     t    10     f   186   4700        +
## 101 583 48.50  4.250  0.125     t     f     0     t   225      0        +
## 102 596 25.75  0.750  0.250     t     f     0     f   349     23        +
## 103 600 20.50  2.415  2.000     t     t    11     t   200   3000        +
## 104 603 29.83  1.250  0.250     f     f     0     f   224      0        -
## 105 616 29.42  1.250  0.250     f     t     2     t   400    108        -
## 106 618 32.25 14.000  0.000     f     t     2     f   160      1        -
## 107 619 29.58  4.750  2.000     f     t     1     t   460     68        -
## 108 627 22.00  7.835  0.165     f     f     0     t    NA      0        -
## 109 630 19.58  0.665  1.665     f     f     0     f   220      5        -
## 110 637 40.92  0.500  0.500     f     f     0     t   130      0        -
## 111 639 28.58  3.625  0.250     f     f     0     t   100      0        -
## 112 641 34.17  2.750  2.500     f     f     0     t   232    200        -
## 113 642 33.17  2.250  3.500     f     f     0     t   200    141        -
## 114 648 24.08  9.000  0.250     f     f     0     t     0      0        -
## 115 651 48.08  3.750  1.000     f     f     0     f   100      2        -
## 116 656 21.08  5.000  0.000     f     f     0     f     0      0        -
## 117 661 22.25  9.000  0.085     f     f     0     f     0      0        -
## 118 665 31.08  1.500  0.040     f     f     0     f   160      0        -
## 119 670 51.83  2.040  1.500     f     f     0     f   120      1        -
```

```
## 120 672 25.83 12.835  0.500     f     f     0     f     0     2        -
## 121 679 17.92 10.210  0.000     f     f     0     f     0    50        -
## 122 685 40.58  3.290  3.500     f     f     0     t   400     0        -
## 123 687 22.67  0.750  2.000     f     t     2     t   200   394        -
## 124   1 30.83  0.000  1.250     t     t     1     f   202     0        +
## 125   2 58.67  4.460  3.040     t     t     6     f    43   560        +
## 126   5 20.17  5.625  1.710     t     f     0     f   120     0        +
## 127   6 32.08  4.000  2.500     t     f     0     t   360     0        +
## 128   7 33.17  1.040  6.500     t     f     0     t   164 31285        +
## 129   8 22.92 11.585  0.040     t     f     0     f    80  1349        +
## 130   9 54.42  0.500  3.960     t     f     0     f   180   314        +
## 131  10 42.50  4.915  3.165     t     f     0     t    52  1442        +
## 132  11 22.08  0.830  2.165     f     f     0     t   128     0        +
## 133  12 29.92  1.835  4.335     t     f     0     f   260   200        +
## 134  13 38.25  6.000  1.000     t     f     0     t     0     0        +
## 135  14 48.08  6.040  0.040     f     f     0     f     0  2690        +
## 136  17 28.25  0.875  0.960     t     t     3     t   396     0        +
## 137  18 23.25  5.875  3.170     t     t    10     f   120   245        +
## 138  21 25.00 11.250  2.500     t     t    17     f   200  1208        +
## 139  22 23.25  1.000  0.835     t     f     0     f   300     0        +
## 140  23 47.75  8.000  7.875     t     t     6     t     0  1260        +
## 141  24 27.42 14.500  3.085     t     t     1     f   120    11        +
## 142  27 47.00 13.000  5.165     t     t     9     t     0     0        +
## 143  29 57.42  8.500  7.000     t     t     3     f     0     0        +
## 144  31 29.25 14.790  5.040     t     t     5     t   168     0        +
## 145  34 36.75  5.125  5.000     t     f     0     t     0  4000        +
## 146  36 27.83  1.500  2.000     t     t    11     t   434    35        +
## 147  37 27.25  1.585  1.835     t     t    12     t   583   713        +
## 148  38 23.00 11.750  0.500     t     t     2     t   300   551        +
## 149  39 27.75  0.585  0.250     t     t     2     f   260   500        +
## 150  42 28.92 15.000  5.335     t     t    11     f     0  2283        +
## 151  43 29.67  1.415  0.750     t     t     1     f   240   100        +
## 152  44 39.58 13.915  8.625     t     t     6     t    70     0        +
## 153  45 56.42 28.000 28.500     t     t    40     f     0    15        +
## 154  49 41.50  1.540  3.500     f     f     0     f   216     0        +
## 155  50 23.92  0.665  0.165     f     f     0     f   100     0        +
## 156  51 25.75  0.500  0.875     t     f     0     t   491     0        +
## 157  52 26.00  1.000  1.750     t     f     0     t   280     0        +
## 158  53 37.42  2.040  0.040     t     f     0     t   400  5800        +
## 159  58 44.33  0.500  5.000     t     f     0     t   320     0        +
## 160  61 56.75 12.250  1.250     t     t     4     t   200     0        +
## 161  66 34.17  1.540  1.540     t     t     1     t   520 50000        +
## 162  67 36.00  1.000  2.000     t     t    11     f     0   456        +
## 163  68 25.50  0.375  0.250     t     t     3     f   260 15108        +
## 164  71 32.33  7.500  1.585     t     f     0     t   420     0        -
## 165  72 34.83  4.000 12.500     t     f     0     t    NA     0        -
## 166  73 38.58  5.000 13.500     t     f     0     t   980     0        -
## 167  74 44.25  0.500 10.750     t     f     0     f   400     0        -
## 168  75 44.83  7.000  1.625     f     f     0     f   160     2        -
## 169  76 20.67  5.290  0.375     t     t     1     f   160     0        -
## 170  81 49.58 19.000  0.000     t     t     1     f    94     0        -
## 171  82 27.67  1.500  2.000     t     f     0     f   368     0        -
## 172  87    NA  0.375  0.875     t     f     0     t   928     0        -
## 173  88 25.67  2.210  4.000     t     f     0     f   188     0        -
```

```
## 174  90 49.00  1.500  0.000     t     f     0     t   100     27        -
## 175  91 62.50 12.750  5.000     t     f     0     f   112      0        -
## 176  93    NA  5.000  8.500     t     f     0     f     0      0        -
## 177  94 52.33  1.375  9.460     t     f     0     t   200    100        -
## 178  98    NA  0.500  0.835     t     f     0     t   320      0        -
## 179 100 28.50  1.000  1.000     t     t     2     t   167    500        -
## 180 101 37.50  1.750  0.250     t     f     0     t   164    400        -
## 181 102 35.25 16.500  4.000     t     f     0     f    80      0        -
## 182 103 18.67  5.000  0.375     t     t     2     f     0     38        -
## 183 105 27.83  4.000  5.750     t     t     2     t    75      0        -
## 184 106 54.83 15.500  0.000     t     t    20     f   152    130        -
## 185 107 28.75  1.165  0.500     t     f     0     f   280      0        -
## 186 113 24.58 12.500  0.875     t     f     0     t   260      0        -
## 187 115 20.67  1.250  1.375     t     t     3     t   140    210        -
## 188 117 37.75  7.000 11.500     t     t     7     t   300      5        -
## 189 118 52.50  6.500  6.290     t     t    15     f     0  11202        +
## 190 119 57.83  7.040 14.000     t     t     6     t   360   1332        +
## 191 122 25.67 12.500  1.210     t     t    67     t   140    258        +
## 192 123 24.75 12.500  1.500     t     t    12     t   120    567        +
## 193 124 44.17  6.665  7.375     t     t     3     t     0      0        +
## 194 125 23.50  9.000  8.500     t     t     5     t   120      0        +
## 195 132 20.42  1.835  2.250     t     t     1     f   100    150        +
## 196 133 47.42  8.000  6.500     t     t     6     f   375  51100        +
## 197 134 36.25  5.000  2.500     t     t     6     f     0    367        +
## 198 135 32.67  5.500  5.500     t     t    12     t   408   1000        +
## 199 137 39.92  0.540  0.500     t     t     3     f   200   1000        +
## 200 139 18.83  9.500  1.625     t     t     6     t    40    600        +
## 201 140 26.92 13.500  5.000     t     t     2     f     0   5000        +
## 202 141 31.25  3.750  0.625     t     t     9     t   181      0        +
## 203 142 56.50 16.000  0.000     t     t    15     f     0    247        +
## 204 143 43.00  0.290  1.750     t     t     8     f   100    375        +
## 205 144 22.33 11.000  2.000     t     t     1     f    80    278        +
## 206 147 23.25  1.500  2.375     t     t     3     t     0    582        +
## 207 148 40.33  7.540  8.000     t     t    14     f     0   2300        +
## 208 149 30.50  6.500  4.000     t     t     7     t     0   3065        +
## 209 150 52.83 15.000  5.500     t     t    14     f     0   2200        +
## 210 151 46.67  0.460  0.415     t     t    11     t   440      6        +
## 211 153 37.33  6.500  4.250     t     t    12     t    93      0        +
## 212 155 32.75  1.500  5.500     t     t     3     t     0      0        +
## 213 157 28.50  3.040  2.540     t     t     1     f    70      0        +
## 214 158 68.67 15.000  0.000     t     t    14     f     0   3376        +
## 215 159 28.00  2.000  4.165     t     t     2     t   181      0        +
## 216 160 34.08  0.080  0.040     t     t     1     t   280   2000        +
## 217 161 27.67  2.000  1.000     t     t     4     f   140   7544        +
## 218 163 25.08  1.710  1.665     t     t     1     t   395     20        +
## 219 165 60.58 16.500 11.000     t     f     0     t    21  10561        +
## 220 167 19.33  9.500  1.000     t     f     0     t    60    400        +
## 221 168 32.33  0.540  0.040     t     f     0     f   440  11177        +
## 222 169 36.67  3.250  9.000     t     f     0     t   102    639        +
## 223 171 25.08  2.540  0.250     t     f     0     t   370      0        +
## 224 172 41.33  0.000 15.000     t     f     0     f     0      0        +
## 225 173 56.00 12.500  8.000     t     f     0     t    24   2028        +
## 226 175 22.67 10.500  1.335     t     f     0     f   100      0        +
## 227 176 27.00  1.500  0.375     t     f     0     t   260   1065        +
```

```
## 228 177 25.00 12.500  3.000     t     f     0     t    20     0        +
## 229 178 26.08  8.665  1.415     t     f     0     f   160   150        +
## 230 179 18.42  9.250  1.210     t     t     4     f    60   540        +
## 231 181 47.67  0.290 15.000     t     t    20     f     0 15000        +
## 232 182 21.25  2.335  0.500     t     t     4     f    80     0        +
## 233 183 20.67  3.000  0.165     t     t     3     f   100     6        +
## 234 185 22.42  5.665  2.585     t     t     7     f   129  3257        +
## 235 187 40.00  6.500  3.500     t     t     1     f     0   500        +
## 236 190 33.08  4.625  1.625     t     t     2     f     0     0        +
## 237 191 21.33 10.500  3.000     t     f     0     t     0     0        +
## 238 192 42.00  0.205  5.125     t     f     0     f   400     0        +
## 239 193 41.75  0.960  2.500     t     f     0     f   510   600        +
## 240 194 22.67  1.585  3.085     t     t     6     f    80     0        +
## 241 196 28.25  5.040  1.500     t     t     8     t   144     7        +
## 242 197 33.17  3.165  3.165     t     t     3     t   380     0        +
## 243 201 24.08  0.500  1.250     t     t     1     f     0   678        +
## 244 202 41.33  1.000  2.250     t     f     0     t     0   300        +
## 245 203 24.83  2.750  2.250     t     t     6     f    NA   600        +
## 246 205 36.33  2.125  0.085     t     t     1     f    50  1187        +
## 247 206 35.42 12.000 14.000     t     t     8     f     0  6590        +
## 248 208 28.67  9.335  5.665     t     t     6     f   381   168        +
## 249 210 39.50  4.250  6.500     t     t    16     f   117  1210        +
## 250 211 39.33  5.875 10.000     t     t    14     t   399     0        +
## 251 213 60.08 14.500 18.000     t     t    15     t     0  1000        +
## 252 214 23.08 11.500  3.500     t     t     9     f    56   742        +
## 253 216 48.17  3.500  3.500     t     f     0     f   230     0        +
## 254 219 53.92  9.625  8.665     t     t     5     f     0     0        +
## 255 220 18.92  9.250  1.000     t     t     4     t    80   500        +
## 256 221 50.08 12.540  2.290     t     t     3     t   156     0        +
## 257 223 17.58  9.000  1.375     t     f     0     t     0     0        +
## 258 224 18.83  9.540  0.085     t     f     0     f   100     0        +
## 259 225 37.75  5.500  0.125     t     f     0     t   228     0        +
## 260 226 23.25  4.000  0.250     t     f     0     t   160     0        +
## 261 227 18.08  5.500  0.500     t     f     0     f    80     0        +
## 262 228 22.50  8.460  2.460     f     f     0     f   164     0        +
## 263 229 19.67  0.375  2.000     t     t     2     t    80     0        +
## 264 231 25.17  3.500  0.625     t     t     7     f     0  7059        +
## 265 232 47.42  3.000 13.875     t     t     2     t   519  1704        +
## 266 233 33.50  1.750  4.500     t     t     4     t   253   857        +
## 267 234 27.67 13.750  5.750     t     f     0     t   487   500        +
## 268 235 58.42 21.000 10.000     t     t    13     f     0  6700        +
## 269 236 20.67  1.835  2.085     t     t     5     f   220  2503        +
## 270 237 26.17  0.250  0.000     t     f     0     t     0     0        +
## 271 238 21.33  7.500  1.415     t     t     1     f    80  9800        +
## 272 240 38.17 10.125  2.500     t     t     6     f   520   196        +
## 273 241 20.50 10.000  2.500     t     f     0     f    40     0        +
## 274 242 48.25 25.085  1.750     t     t     3     f   120    14        +
## 275 244 18.75  7.500  2.710     t     t     5     f    NA 26726        +
## 276 246 33.17  3.040  2.040     t     t     1     t   180 18027        +
## 277 247 45.00  8.500 14.000     t     t     1     t    88  2000        +
## 278 248 19.67  0.210  0.290     t     t    11     f    80    99        +
## 279 249 24.50 12.750  4.750     t     t     2     f    73   444        +
## 280 250 21.83 11.000  0.290     t     t     6     f   121     0        +
## 281 252 41.42  5.000  5.000     t     t     6     t   470     0        +
```

```
## 282 253 17.83 11.000  1.000     t     t    11     f     0   3000        +
## 283 255    NA  0.625  0.250     f     f     0     f   380   2010        -
## 284 256 18.17 10.250  1.085     f     f     0     f   320     13        -
## 285 257 20.00 11.045  2.000     f     f     0     t   136      0        -
## 286 258 20.00  0.000  0.500     f     f     0     f   144      0        -
## 287 266 17.08  0.085  0.040     f     f     0     f   140    722        -
## 288 267 18.33  1.210  0.000     f     f     0     f   100      0        -
## 289 268 32.00  6.000  1.250     f     f     0     f   272      0        -
## 290 269 59.67  1.540  0.125     t     f     0     t   260      0        +
## 291 270 18.00  0.165  0.210     f     f     0     f   200     40        +
## 292 271 37.58  0.000  0.000     f     f     0     f    NA      0        +
## 293 272 32.33  2.500  1.250     f     f     0     t   280      0        -
## 294 273 18.08  6.750  0.040     f     f     0     f   140      0        -
## 295 274 38.25 10.125  0.125     f     f     0     f   160      0        -
## 296 278 18.17 10.000  0.165     f     f     0     f   340      0        -
## 297 280 16.25  0.835  0.085     t     f     0     f   200      0        -
## 298 281 21.17  0.875  0.250     f     f     0     f   280    204        -
## 299 282 23.92  0.585  0.125     f     f     0     f   240      1        -
## 300 283 17.67  4.460  0.250     f     f     0     f    80      0        -
## 301 284 16.50  1.250  0.250     f     t     1     f   108     98        -
## 302 285 23.25 12.625  0.125     f     t     2     f     0   5552        -
## 303 287    NA  1.500  0.000     f     t     2     t   200    105        -
## 304 289 18.83  0.415  0.165     f     t     1     f   200      1        -
## 305 290 21.75  1.750  0.000     f     f     0     f   160      0        -
## 306 291 23.00  0.750  0.500     f     f     0     t   320      0        -
## 307 292 18.25 10.000  1.000     f     t     1     f   120      1        -
## 308 293 25.42  0.540  0.165     f     t     1     f   272    444        -
## 309 297 69.17  9.000  4.000     f     t     1     f    70      6        -
## 310 298 32.92  2.500  1.750     f     t     2     t   720      0        -
## 311 301 57.58  2.000  6.500     f     t     1     f     0     10        -
## 312 302 18.25  0.165  0.250     f     f     0     t   280      0        -
## 313 305 24.75 13.665  1.500     f     f     0     f   280      1        -
## 314 306 48.75 26.335  0.000     t     f     0     t     0      0        -
## 315 309 27.75  1.290  0.250     f     f     0     t   140      0        -
## 316 313 16.33  0.210  0.125     f     f     0     f   200      1        -
## 317 314 18.58 10.000  0.415     f     f     0     f    80     42        -
## 318 315 16.25  0.000  0.250     f     f     0     f    60      0        -
## 319 316 23.00  0.750  0.500     t     f     0     t   320      0        -
## 320 319 19.17  0.000  0.000     f     f     0     t   500      1        +
## 321 320 36.75  0.125  1.500     f     f     0     t   232    113        +
## 322 323 33.67  0.375  0.375     f     f     0     f   300     44        +
## 323 324 48.58  0.205  0.250     t     t    11     f   380   2732        +
## 324 326 29.50  1.085  1.000     f     f     0     f   280     13        -
## 325 327 30.17  1.085  0.040     f     f     0     f   170    179        -
## 326 333 34.08  2.500  1.000     f     f     0     f   460     16        -
## 327 337 47.33  6.500  1.000     f     f     0     t     0    228        -
## 328 339 33.25  3.000  2.000     f     f     0     f   180      0        -
## 329 340 28.00  3.000  0.750     f     f     0     t   300     67        -
## 330 341 39.08  4.000  3.000     f     f     0     f   480      0        -
## 331 342 42.75  4.085  0.040     f     f     0     f   108    100        -
## 332 344 33.75  2.750  0.000     f     f     0     f   180      0        -
## 333 345 38.92  1.750  0.500     f     f     0     t   300      2        -
## 334 347 32.25  1.500  0.250     f     f     0     t   372    122        -
## 335 350 27.83  1.500  2.250     f     t     1     t   100      3        -
```

```
## 336 352 22.17  0.585  0.000     f     f     0     f   100      0        -
## 337 355 36.67  2.000  0.250     f     f     0     t   221      0        -
## 338 357 41.17  1.335  0.165     f     f     0     f   168      0        -
## 339 358 19.50  0.165  0.040     f     f     0     t   380      0        -
## 340 361 30.25  5.500  5.500     f     f     0     t   100      0        -
## 341 362 23.08  2.500  0.085     f     f     0     t   100   4208        -
## 342 363 26.83  0.540  0.000     f     f     0     f   100      0        -
## 343 366 42.83  1.250 13.875     f     t     1     t   352    112        -
## 344 367 22.75  6.165  0.165     f     f     0     f   220   1000        -
## 345 368 39.42  1.710  0.165     f     f     0     f   400      0        -
## 346 370 21.42  0.750  0.750     f     f     0     t   132      2        -
## 347 371 33.00  2.500  7.000     f     f     0     t   280      0        -
## 348 373 45.00  4.585  1.000     f     f     0     t   240      0        -
## 349 374 26.25  1.540  0.125     f     f     0     f   100      0        -
## 350 375 28.17  0.585  0.040     f     f     0     f   260   1004        -
## 351 376 20.83  0.500  1.000     f     f     0     f   260      0        -
## 352 377 28.67 14.500  0.125     f     f     0     f     0    286        -
## 353 379 34.42  1.335  0.125     f     f     0     t   440   4500        -
## 354 380 33.58  0.250  4.000     f     f     0     t   420      0        -
## 355 384 56.83  4.250  5.000     f     f     0     t     0      4        -
## 356 385 22.08 11.460  1.585     f     f     0     t   100   1212        -
## 357 386 34.00  5.500  1.500     f     f     0     t    60      0        -
## 358 387 22.58  1.500  0.540     f     f     0     t   120     67        -
## 359 389 26.67 14.585  0.000     f     f     0     t   178      0        -
## 360 390 22.92  0.170  0.085     f     f     0     f     0      0        -
## 361 391 15.17  7.000  1.000     f     f     0     f   600      0        -
## 362 392 39.92  5.000  0.210     f     f     0     f   550      0        -
## 363 393 27.42 12.500  0.250     f     f     0     t   720      0        -
## 364 394 24.75  0.540  1.000     f     f     0     t   120      1        -
## 365 395 41.17  1.250  0.250     f     f     0     f     0    195        -
## 366 397 29.83  2.040  0.040     f     f     0     f   128      1        -
## 367 399 26.17 12.500  1.250     f     f     0     t     0     17        -
## 368 400 31.00  2.085  0.085     f     f     0     f   300      0        -
## 369 402 28.92  0.375  0.290     f     f     0     f   220    140        -
## 370 404 22.67  0.335  0.750     f     f     0     f   160      0        -
## 371 407 40.33  8.125  0.165     f     t     2     f    NA     18        -
## 372 409 16.00  3.125  0.085     f     t     1     f     0      6        -
## 373 411 31.25  2.835  0.000     f     t     5     f   176    146        -
## 374 412 25.17  3.000  1.250     f     t     1     f     0     22        -
## 375 415 22.25  0.460  0.125     f     f     0     t   280     55        -
## 376 416 22.25  1.250  3.250     f     f     0     f   280      0        -
## 377 417 22.50  0.125  0.125     f     f     0     f   200     70        -
## 378 419 38.42  0.705  0.375     f     t     2     f   225    500        -
## 379 420 26.58  2.540  0.000     f     f     0     t   180     60        -
## 380 421 35.00  2.500  1.000     f     f     0     t   210      0        -
## 381 425 33.67  2.165  1.500     f     f     0     f   120      0        -
## 382 427 27.67  2.040  0.250     f     f     0     t   180     50        -
## 383 428 37.50  0.835  0.040     f     f     0     f   120      5        -
## 384 429 49.17  2.290  0.290     f     f     0     f   200      3        -
## 385 430 33.58  0.335  0.085     f     f     0     f   180      0        -
## 386 432 22.92  3.165  0.165     f     f     0     f   160   1058        -
## 387 434 25.25  1.000  0.500     f     f     0     f   200      0        -
## 388 435 58.58  2.710  2.415     f     f     0     t   320      0        -
## 389 436 19.00  0.000  0.000     f     t     4     f    45      1        -
```

```
## 390 438 53.33  0.165  0.000     f     f     0     t    62     27        -
## 391 439 27.17  1.250  0.000     f     t     1     f    92    300        -
## 392 440 25.92  0.875  0.375     f     t     2     t   174      3        -
## 393 447 16.50  0.125  0.165     f     f     0     f   132      0        -
## 394 448 27.33  1.665  0.000     f     f     0     f   340      1        -
## 395 449 31.25  1.125  0.000     f     t     1     f    96     19        -
## 396 450 20.00  7.000  0.500     f     f     0     f     0      0        -
## 397 452 39.50  1.625  1.500     f     f     0     f     0    316        -
## 398 453 36.50  4.250  3.500     f     f     0     f   454     50        -
## 399 455 52.42  1.500  3.750     f     f     0     t     0    350        -
## 400 456 36.17 18.125  0.085     f     f     0     f   320   3552        -
## 401 457 34.58  0.000  0.000     f     f     0     f    NA      0        -
## 402 459 36.17  5.500  5.000     f     f     0     f   210    687        -
## 403 460 25.67  0.290  1.500     f     f     0     t   160      0        -
## 404 462 24.08  0.875  0.085     f     t     4     f   254   1950        -
## 405 463 21.92  0.500  0.125     f     f     0     f   360      0        -
## 406 465 23.00  1.835  0.000     f     t     1     f   200     53        -
## 407 467 31.08  3.085  2.500     f     t     2     t   160     41        -
## 408 468 30.42  1.375  0.040     f     t     3     f     0     33        -
## 409 470 16.33  4.085  0.415     f     f     0     t   120      0        -
## 410 473 17.42  6.500  0.125     f     f     0     f    60    100        -
## 411 474 19.17  4.000  1.000     f     f     0     t   360   1000        -
## 412 476 26.75  2.000  0.750     f     f     0     t    80      0        -
## 413 477 23.58  0.835  0.085     f     f     0     t   220      5        -
## 414 479 22.75 11.500  0.415     f     f     0     f     0      0        -
## 415 481 16.92  0.500  0.165     f     t     6     t   240     35        -
## 416 482 23.50  3.165  0.415     f     t     1     t   280     80        -
## 417 483 17.33  9.500  1.750     f     t    10     t     0     10        -
## 418 484 23.75  0.415  0.040     f     t     2     f   128      6        -
## 419 485 34.67  1.080  1.165     f     f     0     f    28      0        -
## 420 487 28.17  0.125  0.085     f     f     0     f   216   2100        -
## 421 489 18.83  3.540  0.000     f     f     0     t   180      1        -
## 422 492 24.17  0.875  4.625     t     t     2     t   520   2000        +
## 423 493 39.25  9.500  6.500     t     t    14     f   240   4607        +
## 424 496 19.17  9.500  1.500     t     f     0     f   120   2206        +
## 425 497 25.00  0.875  1.040     t     f     0     t   160   5860        +
## 426 500 20.42  7.000  1.625     t     t     3     f   200   1391        +
## 427 501    NA  4.000  5.000     t     t     3     t   290   2279        +
## 428 502 39.00  5.000  3.500     t     t    10     t     0      0        +
## 429 503 64.08  0.165  0.000     t     t     1     f   232    100        +
## 430 504 28.25  5.125  4.750     t     t     2     f   420      7        +
## 431 505 28.75  3.750  1.085     t     t     1     t   371      0        +
## 432 506 31.33 19.500  7.000     t     t    16     f     0   5000        +
## 433 507 18.92  9.000  0.750     t     t     2     f    88    591        +
## 434 508 24.75  3.000  1.835     t     t    19     f     0    500        +
## 435 509 30.67 12.000  2.000     t     t     1     f   220     19        +
## 436 510 21.00  4.790  2.250     t     t     1     t    80    300        +
## 437 513 44.33  0.000  2.500     t     f     0     f     0      0        +
## 438 514 20.25  9.960  0.000     t     f     0     f     0      0        +
## 439 515 22.67  2.540  2.585     t     f     0     f     0      0        +
## 440 516    NA 10.500  6.500     t     f     0     f     0      0        +
## 441 518 16.08  0.750  1.750     t     t     5     t   352    690        +
## 442 521 20.42  7.500  1.500     t     t     1     f   160    234        +
## 443 522 30.00  5.290  2.250     t     t     5     t    99    500        +
```

```
## 444 525 28.58  1.665  2.415    t    f    0    t   440     0       -
## 445 526 45.17  1.500  2.500    t    f    0    t   140     0       -
## 446 527 41.58  1.750  0.210    t    f    0    f   160     0       -
## 447 528 57.08  0.335  1.000    t    f    0    t   252  2197       -
## 448 529 55.75  7.080  6.750    t    t    3    t   100    50       -
## 449 530 43.25 25.210  0.210    t    t    1    f   760    90       -
## 450 532 24.58  0.670  1.750    t    f    0    f   400     0       -
## 451 533 43.17  2.250  0.750    t    f    0    f   560     0       -
## 452 536 33.92  1.585  0.000    t    f    0    f   320     0       -
## 453 537 24.92  1.250  0.000    t    f    0    f    80     0       -
## 454 539 34.25  1.750  0.250    t    f    0    t   163     0       -
## 455 540 80.25  5.500  0.540    t    f    0    f     0   340       -
## 456 541 19.42  1.500  2.000    t    f    0    t   100    20       -
## 457 544 36.33  3.790  1.165    t    f    0    t   200     0       -
## 458 546 44.25 11.000  1.500    t    f    0    f     0     0       -
## 459 547 23.58  0.460  2.625    t    t    6    t   208   347       -
## 460 548 23.92  1.500  1.875    t    t    6    f   200   327       +
## 461 549 33.17  1.000  0.750    t    t    7    t   340  4071       +
## 462 552 51.33 10.000  0.000    t    t   11    f     0  1249       +
## 463 554 38.58  3.335  4.000    t    t   14    f   383  1344       +
## 464 556 41.92  0.420  0.210    t    t    6    f   220   948       +
## 465 558 32.17  1.460  1.085    t    t   16    f   120  2079       +
## 466 559 51.42  0.040  0.040    t    f    0    f     0  3000       +
## 467 560 22.83  2.290  2.290    t    t    7    t   140  2384       +
## 468 563 23.33  1.500  1.415    t    f    0    f   422   200       +
## 469 564 24.42 12.335  1.585    t    f    0    t   120     0       +
## 470 565 42.17  5.040 12.750    t    f    0    t    92     0       +
## 471 568 25.17  2.875  0.875    t    f    0    f   360     0       +
## 472 569 43.08  0.375  0.375    t    t    8    t   300   162       +
## 473 570 35.75  0.915  0.750    t    t    4    f     0  1583       +
## 474 571 59.50  2.750  1.750    t    t    5    t    60    58       +
## 475 573 21.92  0.540  0.040    t    t    1    t   840    59       +
## 476 577 30.17  0.500  1.750    t    t   11    f    32   540       +
## 477 578 25.17  6.000  1.000    t    t    3    f     0     0       +
## 478 581 31.67  0.830  1.335    t    t    8    t   303  3290       +
## 479 584 32.67  9.000  5.250    t    f    0    t   154     0       +
## 480 585 28.08 15.000  0.000    t    f    0    f     0 13212       +
## 481 587 64.08 20.000 17.500    t    t    9    t     0  1000       +
## 482 588 51.58 15.000  8.500    t    t    9    f     0     0       +
## 483 589 26.67  1.750  1.000    t    t    5    t   160  5777       +
## 484 591 30.17  6.500  3.125    t    t    8    f   330  1200       +
## 485 592 27.00  0.750  4.250    t    t    3    t   312   150       +
## 486 593 23.17  0.000  0.000    f    f    0    f    NA     0       +
## 487 594 34.17  5.250  0.085    f    f    0    t   290     6       +
## 488 595 38.67  0.210  0.085    t    f    0    t   280     0       +
## 489 597 46.08  3.000  2.375    t    t    8    t   396  4159       +
## 490 598 21.50  6.000  2.500    t    t    3    f    80   918       +
## 491 599 20.08  0.125  1.000    f    t    1    f   240   768       +
## 492 601 29.50  0.460  0.540    t    t    4    f   380   500       +
## 493 604 20.08  0.250  0.125    f    f    0    f   200     0       -
## 494 605 23.42  0.585  0.085    t    f    0    f   180     0       -
## 495 606 29.58  1.750  1.250    f    f    0    t   280     0       -
## 496 607 16.17  0.040  0.040    f    f    0    f     0     0       +
## 497 608 32.33  3.500  0.500    f    f    0    t   232     0       -
```

```
## 498 611 20.00  1.250  0.125    f    f    0    f   140     4       -
## 499 612 27.58  3.250  5.085    f    t    2    t   369     1       -
## 500 613 22.00  0.790  0.290    f    t    1    f   420   283       -
## 501 614 19.33 10.915  0.585    f    t    2    t   200     7       -
## 502 617 22.67  0.750  1.585    f    t    1    t   400     9       -
## 503 620 18.42 10.415  0.125    t    f    0    f   120   375       -
## 504 621 22.17  2.250  0.125    f    f    0    f   160    10       -
## 505 622 22.67  0.165  2.250    f    f    0    t     0     0       +
## 506 623 25.58  0.000  0.000    f    f    0    f    NA     0       +
## 507 624 18.83  0.000  0.665    f    f    0    f   160     1       -
## 508 625 21.58  0.790  0.665    f    f    0    f   160     0       -
## 509 626 23.75 12.000  2.085    f    f    0    f    80     0       -
## 510 628 36.08  2.540  0.000    f    f    0    f     0  1000       -
## 511 631 22.92  1.250  0.250    f    f    0    t   120   809       -
## 512 633 38.75  1.500  0.000    f    f    0    f    76     0       -
## 513 634 32.42  2.165  0.000    f    f    0    f   120     0       -
## 514 635 23.75  0.710  0.250    f    t    1    t   240     4       -
## 515 636 18.17  2.460  0.960    f    t    2    t   160   587       -
## 516 638 19.50  9.585  0.790    f    f    0    f    80   350       -
## 517 640 35.58  0.750  1.500    f    f    0    t   231     0       -
## 518 643 31.58  0.750  3.500    f    f    0    t   320     0       -
## 519 645 36.17  0.420  0.290    f    f    0    t   309     2       -
## 520 646 37.33  2.665  0.165    f    f    0    t     0   501       -
## 521 647 20.83  8.500  0.165    f    f    0    f     0   351       -
## 522 652 15.83  7.625  0.125    f    t    1    t     0   160       -
## 523 653 22.50  0.415  0.335    f    f    0    t   144     0       -
## 524 655 23.58  0.830  0.415    f    t    1    t   200    11       -
## 525 657 25.67  3.250  2.290    f    t    1    t   416    21       -
## 526 658 38.92  1.665  0.250    f    f    0    f     0   390       -
## 527 659 15.75  0.375  1.000    f    f    0    f   120    18       -
## 528 660 28.58  3.750  0.250    f    t    1    t    40   154       -
## 529 663 23.50  1.500  0.875    f    f    0    t   160     0       -
## 530 664 32.08  4.000  1.500    f    f    0    t   120     0       -
## 531 668 17.92  0.540  1.750    f    t    1    t    80     5       -
## 532 669 30.33  0.500  0.085    f    f    0    t   252     0       -
## 533 673 50.25  0.835  0.500    f    f    0    t   240   117       -
## 534 674 29.50  2.000  2.000    f    f    0    f   256    17       -
## 535 675 37.33  2.500  0.210    f    f    0    f   260   246       -
## 536 676 41.58  1.040  0.665    f    f    0    f   240   237       -
## 537 678 19.42  7.250  0.040    f    t    1    f   100     1       -
## 538 681 19.50  0.290  0.290    f    f    0    f   280   364       -
## 539 684 36.42  0.750  0.585    f    f    0    f   240     3       -
## 540 688 25.25 13.500  2.000    f    t    1    t   200     1       -
## 541 689 17.92  0.205  0.040    f    f    0    f   280   750       -
## 542  15 45.83 10.500  5.000    t    t    7    t     0     0       +
## 543  20 19.17  8.585  0.750    t    t    7    f    96     0       +
## 544  25 41.17  6.500  0.500    t    t    3    t   145     0       +
## 545  30 42.08  1.040  5.000    t    t    6    t   500 10000       +
## 546  32 42.00  9.790  7.960    t    t    8    f     0     0       +
## 547  35 22.58 10.750  0.415    t    t    5    t     0   560       +
## 548  48 31.92  4.460  6.040    t    t    3    f   311   300       +
## 549  54 34.92  2.500  0.000    t    f    0    t   239   200       +
## 550  55 34.25  3.000  7.415    t    f    0    t     0     0       +
## 551  57 23.17  0.000  0.085    t    f    0    f     0     0       +
```

```
## 552  60 43.25  3.000  6.000     t     t    11     f    80      0        +
## 553  62 31.67 16.165  3.000     t     t     9     f   250    730        +
## 554  64 20.42  0.835  1.585     t     t     1     f     0      0        +
## 555  70 35.17 25.125  1.625     t     t     1     t   515    500        +
## 556  78 19.17  0.585  0.585     t     f     0     t   160      0        -
## 557  79 21.67  1.165  2.500     t     t     1     f   180     20        -
## 558  83 39.83  0.500  0.250     t     f     0     f   288      0        -
## 559  84    NA  3.500  3.000     t     f     0     t   300      0        -
## 560  85 27.25  0.625  0.455     t     f     0     t   200      0        -
## 561  86 37.17  4.000  5.000     t     f     0     t   280      0        -
## 562  89 34.00  4.500  1.000     t     f     0     t   240      0        -
## 563  92 31.42 15.500  0.500     t     f     0     f   120      0        -
## 564  95 28.75  1.500  1.500     t     f     0     t     0    225        -
## 565  96 28.58  3.540  0.500     t     f     0     t   171      0        -
## 566  97 23.00  0.625  0.125     t     f     0     f   180      1        -
## 567 108 25.00 11.000  4.500     t     f     0     f   120      0        -
## 568 109 40.92  2.250 10.000     t     f     0     t   176      0        -
## 569 111 29.17  3.500  3.500     t     t     3     t   329      0        -
## 570 112 24.50  1.040  0.500     t     t     3     f   180    147        -
## 571 114 33.75  0.750  1.000     t     t     3     t   212      0        -
## 572 116 25.42  1.125  1.290     t     t     2     f   200      0        -
## 573 120 20.75 10.335  0.335     t     t     1     t    80     50        +
## 574 126 34.92  5.000  7.500     t     t     6     t     0   1000        +
## 575 127 47.67  2.500  2.500     t     t    12     t   410   2510        +
## 576 128 22.75 11.000  2.500     t     t     7     t   100    809        +
## 577 131 67.75  5.500 13.000     t     t     1     t     0      0        +
## 578 136 48.58  6.500  6.000     t     f     0     t   350      0        +
## 579 145 27.25  1.665  5.085     t     t     9     f   399    827        +
## 580 152 58.33 10.000  4.000     t     t    14     f     0   1602        +
## 581 156 21.67 11.500  0.000     t     t    11     t     0      0        +
## 582 164 32.00  1.750  0.040     t     f     0     t   393      0        +
## 583 166 40.83 10.000  1.750     t     f     0     f    29    837        +
## 584 170 37.50  1.125  1.500     f     f     0     t   431      0        +
## 585 174 49.83 13.585  8.500     t     f     0     t     0      0        +
## 586 180 20.17  8.170  1.960     t     t    14     f    60    158        +
## 587 184 57.08 19.500  5.500     t     t     7     f     0   3000        +
## 588 186 48.75  8.500 12.500     t     t     9     f   181   1655        +
## 589 195 34.50  4.040  8.500     t     t     7     t   195      0        +
## 590 198 48.17  7.625 15.500     t     t    12     f     0    790        +
## 591 200 22.58 10.040  0.040     t     t     9     f    60    396        +
## 592 204 20.75 10.250  0.710     t     t     2     t    49      0        +
## 593 215 26.67  2.710  5.250     t     t     1     f   211      0        +
## 594 217 41.17  4.040  7.000     t     t     8     f   320      0        +
## 595 218 55.92 11.500  5.000     t     t     5     f     0   8851        +
## 596 222 65.42 11.000 20.000     t     t     7     t    22      0        +
## 597 230 22.08 11.000  0.665     t     f     0     f   100      0        +
## 598 243 28.33  5.000 11.000     t     f     0     t    70      0        +
## 599 245 18.50  2.000  1.500     t     t     2     f   120    300        +
## 600 251 40.25 21.500 20.000     t     t    11     f     0   1200        +
## 601 259 20.75  9.540  0.040     f     f     0     f   200   1000        -
## 602 260 24.50  1.750  0.165     f     f     0     f   132      0        -
## 603 261 32.75  2.335  5.750     f     f     0     t   292      0        -
## 604 262 52.17  0.000  0.000     f     f     0     f     0      0        -
## 605 264 20.42 10.500  0.000     f     f     0     t   154     32        -
```

```
## 606 279 24.58 13.500  0.000     f     f     0     f    NA      0        -
## 607 295 16.08  0.335  0.000     f     t     1     f   160    126        -
## 608 296 31.92  3.125  3.040     f     t     2     t   200      4        -
## 609 299 16.33  2.750  0.665     f     t     1     f    80     21        -
## 610 303 23.42  1.000  0.500     f     f     0     t   280      0        -
## 611 307 23.50  2.750  4.500     f     f     0     f   160     25        -
## 612 308 18.58 10.290  0.415     f     f     0     f    80      0        -
## 613 310 31.75  3.000  0.000     f     f     0     f   160     20        -
## 614 317 21.17  0.250  0.250     f     f     0     f   280    204        -
## 615 321 21.25  1.500  1.500     f     f     0     f   150      8        +
## 616 322 18.08  0.375 10.000     f     f     0     t   300      0        +
## 617 328 40.83  3.500  0.500     f     f     0     f  1160      0        -
## 618 338 34.83  1.250  0.500     f     f     0     t   160      0        -
## 619 343 26.92  2.250  0.500     f     f     0     t   640   4000        -
## 620 346 62.75  7.000  0.000     f     f     0     f     0     12        -
## 621 353 22.50 11.500  1.500     f     f     0     t     0   4000        -
## 622 356 16.00  0.165  1.000     f     t     2     t   320      1        -
## 623 360 36.75  4.710  0.000     f     f     0     f   160      0        -
## 624 365 24.42  2.000  0.165     f     t     2     f   320   1300        -
## 625 369 23.58 11.500  3.000     f     f     0     t    20     16        -
## 626 378 20.67  0.835  2.000     f     f     0     t   240      0        -
## 627 381 43.17  5.000  2.250     f     f     0     t   141      0        -
## 628 382 22.67  7.000  0.165     f     f     0     f   160      0        -
## 629 401 20.75  5.085  0.290     f     f     0     f   140    184        -
## 630 405 34.00  5.085  1.085     f     f     0     t   480      0        -
## 631 406 69.50  6.000  0.000     f     f     0     f     0      0        -
## 632 408 19.58  0.665  1.000     f     t     1     f  2000      2        -
## 633 410 17.08  0.250  0.335     f     t     4     f   160      8        -
## 634 413 22.67  0.790  0.085     f     f     0     f   144      0        -
## 635 414 40.58  1.500  0.000     f     f     0     f   300      0        -
## 636 433 21.83  1.540  0.085     f     f     0     t   356      0        -
## 637 437 19.58  0.585  0.000     f     t     3     f   350    769        -
## 638 441 23.08  0.000  1.000     f     t    11     f     0      0        -
## 639 442 39.58  5.000  0.000     f     t     2     f    17      1        -
## 640 445 17.67  0.000  0.000     f     f     0     f    86      0        -
## 641 446    NA 11.250  0.000     f     f     0     f    NA   5200        -
## 642 451    NA  3.000  7.000     f     f     0     f     0      1        -
## 643 454 29.75  0.665  0.250     f     f     0     t   300      0        -
## 644 458 29.67  0.750  0.040     f     f     0     f   240      0        -
## 645 464 36.58  0.290  0.000     f     t    10     f   200     18        -
## 646 466 27.58  3.000  2.790     f     t     1     t   280     10        -
## 647 475 20.67  0.415  0.125     f     f     0     f     0     44        -
## 648 480 26.50  2.710  0.085     f     f     0     f    80      0        -
## 649 486 74.83 19.000  0.040     f     t     2     f     0    351        -
## 650 488 24.50 13.335  0.040     f     f     0     t   120    475        -
## 651 490 45.33  1.000  0.125     f     f     0     t   263      0        -
## 652 494 20.50 11.835  6.000     t     f     0     f   340      0        +
## 653 495 18.83  4.415  3.000     t     f     0     f   240      0        +
## 654 498 20.17  9.250  1.665     t     t     3     t    40     28        +
## 655 517 60.92  5.000  4.000     t     t     4     f     0     99        +
## 656 519 28.17  0.375  0.585     t     t     4     f    80      0        +
## 657 524 22.50  8.500  1.750     t     t    10     f    80    990        -
## 658 534 40.92  0.835  0.000     t     f     0     f   130      1        -
## 659 545 30.08  1.040  0.500     t     t    10     t   132     28        -
```

```
## 660 550 48.33 12.000 16.000     t     f     0     f   110      0        +
## 661 555 22.42 11.250  0.750     t     t     4     f     0    321        +
## 662 557 29.58  4.500  7.500     t     t     2     t   330      0        +
## 663 561 25.00 12.330  3.500     t     t     6     f   400    458        +
## 664 572 21.00  3.000  1.085     t     t     8     t   160      1        +
## 665 574 65.17 14.000  0.000     t     t    11     t     0   1400        +
## 666 576 32.25  0.165  3.250     t     t     1     t   432   8000        +
## 667 580 39.08  6.000  1.290     t     t     5     t   108   1097        +
## 668 582 41.00  0.040  0.040     f     t     1     f   560      0        +
## 669 586 73.42 17.750  0.000     t     f     0     t     0      0        +
## 670 590 25.33  0.580  0.290     t     t     7     t    96   5124        +
## 671 602 42.25  1.750  0.000     f     f     0     t   150      1        -
## 672 609    NA  0.040  4.250     f     f     0     t   460      0        -
## 673 610 47.83  4.165  0.085     f     f     0     t   520      0        -
## 674 615 38.33  4.415  0.125     f     f     0     f   160      0        -
## 675 629 29.25 13.000  0.500     f     f     0     f   228      0        -
## 676 632 27.25  0.290  0.125     f     t     1     t   272    108        -
## 677 644 52.50  7.000  3.000     f     f     0     f     0      0        -
## 678 649 25.58  0.335  3.500     f     f     0     t   340      0        -
## 679 650 35.17  3.750  0.000     f     t     6     f     0    200        -
## 680 654 21.50 11.500  0.500     t     f     0     t   100     68        -
## 681 662 29.83  3.500  0.165     f     f     0     f   216      0        -
## 682 666 31.83  0.040  0.040     f     f     0     f     0      0        -
## 683 667 21.75 11.750  0.250     f     f     0     t   180      0        -
## 684 671 47.17  5.835  5.500     f     f     0     f   465    150        -
## 685 677 30.58 10.665  0.085     f     t    12     t   129      3        -
## 686 680 20.08  1.250  0.000     f     f     0     f     0      0        -
## 687 682 27.83  1.000  3.000     f     f     0     f   176    537        -
## 688 683 17.08  3.290  0.335     f     f     0     t   140      2        -
## 689 686 21.08 10.085  1.250     f     f     0     f   260      0        -
## 690 690 35.00  3.375  8.290     f     f     0     t     0      0        -
##     credit.score ages ages_zscore credit.score_decimal cont6_minmax
## 1        634.8882   29 -0.90956355               0.62182      0.00824
## 2        634.8882   58  1.56163355               0.65397      0.00003
## 3        634.8882   53  1.13556508               0.60064      0.00000
## 4        634.8882   65  2.15812940               0.65296      0.00000
## 5        634.8882   46  0.53906923               0.59082      0.00000
## 6        634.8882   41  0.11300077               0.62393      0.00000
## 7        634.8882   52  1.05035139               0.65367      0.05000
## 8        634.8882   46  0.53906923               0.65345      0.00300
## 9        634.8882   64  2.07291571               0.62875      0.00221
## 10       634.8882   54  1.22077878               0.65399      0.00284
## 11       634.8882   42  0.19821446               0.63595      0.01236
## 12       634.8882   51  0.96513770               0.63310      0.00300
## 13       634.8882   52  1.05035139               0.64796      0.00000
## 14       634.8882   50  0.87992400               0.63555      0.00400
## 15       634.8882   38 -0.14264031               0.63783      0.00000
## 16       634.8882   43  0.28342815               0.63259      0.02954
## 17       634.8882   63  1.98770201               0.65594      0.00000
## 18       634.8882   62  1.90248832               0.65755      0.00000
## 19       634.8882   53  1.13556508               0.60908      0.00000
## 20       634.8882   54  1.22077878               0.65048      0.00005
## 21       634.8882   36 -0.31306770               0.62449      0.00005
## 22       634.8882   47  0.62428292               0.60925      0.00300
```

```
## 23      634.8882   45  0.45385554          0.64512      0.00610
## 24      634.8882   57  1.47641985          0.62472      0.00000
## 25      634.8882   50  0.87992400          0.63224      0.00000
## 26      634.8882   53  1.13556508          0.63693      0.02072
## 27      634.8882   66  2.24334309          0.65472      0.02184
## 28      634.8882   60  1.73206093          0.62056      0.00015
## 29      634.8882   40  0.02778707          0.64645      0.03065
## 30      634.8882   53  1.13556508          0.65409      0.01430
## 31      634.8882   53  1.13556508          0.63239      0.00560
## 32      634.8882   40  0.02778707          0.64730      0.00000
## 33      634.8882   42  0.19821446          0.61935      0.01270
## 34      634.8882   50  0.87992400          0.61353      0.00000
## 35      634.8882   34 -0.48349508          0.60879      0.00000
## 36      634.8882   33 -0.56870878          0.63378      0.00000
## 37      634.8882   32 -0.65392247          0.63842      0.00120
## 38      634.8882   24 -1.33563201          0.60225      0.00000
## 39      634.8882   32 -0.65392247          0.62906      0.00000
## 40      634.8882   28 -0.99477724          0.62301      0.00000
## 41      634.8882   34 -0.48349508          0.63616      0.00484
## 42      634.8882   32 -0.65392247          0.63993      0.00001
## 43      634.8882   17 -1.93212787          0.64093      0.02803
## 44      634.8882   31 -0.73913616          0.63599      0.00001
## 45      634.8882   25 -1.25041832          0.64353      0.00173
## 46      634.8882   24 -1.33563201          0.64766      0.00000
## 47      634.8882   25 -1.25041832          0.64394      0.00006
## 48      634.8882   27 -1.07999094          0.63840      0.00006
## 49      634.8882   29 -0.90956355          0.64013      1.00000
## 50      634.8882   36 -0.31306770          0.64563      0.00000
## 51      634.8882   36 -0.31306770          0.65234      0.00000
## 52      634.8882   29 -0.90956355          0.65367      0.00000
## 53      634.8882   32 -0.65392247          0.63027      0.00000
## 54      634.8882   23 -1.42084571          0.61317      0.00002
## 55      634.8882   19 -1.76170048          0.64733      0.01062
## 56      634.8882   26 -1.16520463          0.65421      0.00000
## 57      634.8882   41  0.11300077          0.65639      0.00251
## 58      634.8882   25 -1.25041832          0.64413      0.01210
## 59      634.8882   34 -0.48349508          0.65402      0.00000
## 60      634.8882   37 -0.22785401          0.64850      0.00001
## 61      634.8882   32 -0.65392247          0.65576      0.00000
## 62      634.8882   31 -0.73913616          0.62084      0.00000
## 63      634.8882   25 -1.25041832          0.62957      0.00000
## 64      634.8882   38 -0.14264031          0.58514      0.01110
## 65      634.8882   31 -0.73913616          0.61894      0.00456
## 66      634.8882   27 -1.07999094          0.63171      0.00000
## 67      634.8882   28 -0.99477724          0.65294      0.00000
## 68      634.8882   20 -1.67648679          0.64759      0.00087
## 69      634.8882   27 -1.07999094          0.64369      0.00000
## 70      634.8882   38 -0.14264031          0.62521      0.00001
## 71      634.8882   23 -1.42084571          0.58938      0.00007
## 72      634.8882   24 -1.33563201          0.64699      0.00000
## 73      634.8882   35 -0.39828139          0.64455      0.00000
## 74      634.8882   26 -1.16520463          0.62327      0.00000
## 75      634.8882   26 -1.16520463          0.64338      0.00004
## 76      634.8882   28 -0.99477724          0.64906      0.00000
```

```
## 77      634.8882   37 -0.22785401          0.64684      0.00040
## 78      634.8882   36 -0.31306770          0.62815      0.00000
## 79      634.8882   38 -0.14264031          0.63125      0.00000
## 80      634.8882   38 -0.14264031          0.64083      0.00005
## 81      634.8882   21 -1.59127309          0.65312      0.00100
## 82      634.8882   29 -0.90956355          0.63416      0.00000
## 83      634.8882   30 -0.82434986          0.60868      0.00892
## 84      634.8882   45  0.45385554          0.63035      0.00000
## 85      634.8882   30 -0.82434986          0.63346      0.01000
## 86      634.8882   38 -0.14264031          0.65188      0.00960
## 87      634.8882   49  0.79471031          0.60021      0.00000
## 88      634.8882   40  0.02778707          0.65067      0.00800
## 89      634.8882   45  0.45385554          0.65377      0.00001
## 90      634.8882   35 -0.39828139          0.63290      0.00000
## 91      634.8882   41  0.11300077          0.61955      0.00000
## 92      634.8882   30 -0.82434986          0.62093      0.00200
## 93      634.8882   43  0.28342815          0.64127      0.00000
## 94      634.8882   27 -1.07999094          0.61821      0.00109
## 95      634.8882   32 -0.65392247          0.65262      0.00134
## 96      634.8882   32 -0.65392247          0.65544      0.05298
## 97      634.8882   40  0.02778707          0.64233      0.00000
## 98      634.8882   42  0.19821446          0.64975      0.00284
## 99      634.8882   37 -0.22785401          0.62707      0.01465
## 100     634.8882   32 -0.65392247          0.63385      0.04700
## 101     634.8882   45  0.45385554          0.62941      0.00000
## 102     634.8882   31 -0.73913616          0.65695      0.00023
## 103     634.8882   45  0.45385554          0.64600      0.03000
## 104     634.8882   42  0.19821446          0.63532      0.00000
## 105     634.8882   41  0.11300077          0.62905      0.00108
## 106     634.8882   53  1.13556508          0.60717      0.00001
## 107     634.8882   29 -0.90956355          0.64992      0.00068
## 108     634.8882   36 -0.31306770          0.58366      0.00000
## 109     634.8882   44  0.36864185          0.64824      0.00005
## 110     634.8882   47  0.62428292          0.64936      0.00000
## 111     634.8882   41  0.11300077          0.62558      0.00000
## 112     634.8882   37 -0.22785401          0.63340      0.00200
## 113     634.8882   41  0.11300077          0.59178      0.00141
## 114     634.8882   41  0.11300077          0.65546      0.00000
## 115     634.8882   39 -0.05742662          0.60194      0.00002
## 116     634.8882   34 -0.48349508          0.65192      0.00000
## 117     634.8882   37 -0.22785401          0.63708      0.00000
## 118     634.8882   38 -0.14264031          0.64480      0.00000
## 119     634.8882   42  0.19821446          0.63572      0.00001
## 120     634.8882   25 -1.25041832          0.59229      0.00002
## 121     634.8882   55  1.30599247          0.62838      0.00050
## 122     634.8882   44  0.36864185          0.65614      0.00000
## 123     634.8882   46  0.53906923          0.64356      0.00394
## 124     694.1668   42  0.19821446          0.66460      0.00000
## 125     694.1668   54  1.22077878          0.69388      0.00560
## 126     694.1668   65  2.15812940          0.67026      0.00000
## 127     694.1668   61  1.81727463          0.67216      0.00000
## 128     694.1668   50  0.87992400          0.69353      0.31285
## 129     694.1668   41  0.11300077          0.69757      0.01349
## 130     694.1668   30 -0.82434986          0.70422      0.00314
```

```
## 131     694.1668   35 -0.39828139          0.72297      0.01442
## 132     694.1668   57  1.47641985          0.67764      0.00000
## 133     694.1668   58  1.56163355          0.67562      0.00200
## 134     694.1668   60  1.73206093          0.65788      0.00000
## 135     694.1668   43  0.28342815          0.71700      0.02690
## 136     694.1668   40  0.02778707          0.71213      0.00000
## 137     694.1668   48  0.70949662          0.68223      0.00245
## 138     694.1668   38 -0.14264031          0.66453      0.01208
## 139     694.1668   51  0.96513770          0.71016      0.00000
## 140     694.1668   56  1.39120616          0.69552      0.01260
## 141     694.1668   48  0.70949662          0.66291      0.00011
## 142     694.1668   51  0.96513770          0.67787      0.00000
## 143     694.1668   64  2.07291571          0.67594      0.00000
## 144     694.1668   56  1.39120616          0.70556      0.00000
## 145     694.1668   54  1.22077878          0.66327      0.04000
## 146     694.1668   53  1.13556508          0.66491      0.00035
## 147     694.1668   30 -0.82434986          0.68070      0.00713
## 148     694.1668   39 -0.05742662          0.73097      0.00551
## 149     694.1668   67  2.32855678          0.66098      0.00500
## 150     694.1668   20 -1.67648679          0.69767      0.02283
## 151     694.1668   56  1.39120616          0.71706      0.00100
## 152     694.1668   51  0.96513770          0.69727      0.00000
## 153     694.1668   56  1.39120616          0.69762      0.00015
## 154     694.1668   57  1.47641985          0.71299      0.00000
## 155     694.1668   59  1.64684724          0.68482      0.00000
## 156     694.1668   43  0.28342815          0.71912      0.00000
## 157     694.1668   45  0.45385554          0.72722      0.00000
## 158     694.1668   26 -1.16520463          0.71425      0.05800
## 159     694.1668   48  0.70949662          0.68192      0.00000
## 160     694.1668   54  1.22077878          0.70503      0.00000
## 161     694.1668   50  0.87992400          0.69023      0.50000
## 162     694.1668   28 -0.99477724          0.68744      0.00456
## 163     694.1668   39 -0.05742662          0.70975      0.15108
## 164     694.1668   32 -0.65392247          0.65794      0.00000
## 165     694.1668   53  1.13556508          0.67426      0.00000
## 166     694.1668   61  1.81727463          0.68936      0.00000
## 167     694.1668   29 -0.90956355          0.72365      0.00000
## 168     694.1668   48  0.70949662          0.71443      0.00002
## 169     694.1668   67  2.32855678          0.69365      0.00000
## 170     694.1668   40  0.02778707          0.70586      0.00000
## 171     694.1668   68  2.41377048          0.71399      0.00000
## 172     694.1668   30 -0.82434986          0.67758      0.00000
## 173     694.1668   55  1.30599247          0.66699      0.00000
## 174     694.1668   47  0.62428292          0.65990      0.00027
## 175     694.1668   66  2.24334309          0.67247      0.00000
## 176     694.1668   53  1.13556508          0.69988      0.00000
## 177     694.1668   39 -0.05742662          0.68746      0.00100
## 178     694.1668   60  1.73206093          0.72307      0.00000
## 179     694.1668   50  0.87992400          0.73163      0.00500
## 180     694.1668   57  1.47641985          0.66584      0.00400
## 181     694.1668   47  0.62428292          0.67301      0.00000
## 182     694.1668   56  1.39120616          0.67856      0.00038
## 183     694.1668   57  1.47641985          0.69204      0.00000
## 184     694.1668   49  0.79471031          0.72412      0.00130
```

```
## 185     694.1668   46  0.53906923          0.65913      0.00000
## 186     694.1668   39 -0.05742662          0.72587      0.00000
## 187     694.1668   58  1.56163355          0.72082      0.00210
## 188     694.1668   51  0.96513770          0.66690      0.00005
## 189     694.1668   65  2.15812940          0.68909      0.11202
## 190     694.1668   52  1.05035139          0.67887      0.01332
## 191     694.1668   47  0.62428292          0.66626      0.00258
## 192     694.1668   37 -0.22785401          0.70550      0.00567
## 193     694.1668   43  0.28342815          0.66153      0.00000
## 194     694.1668   49  0.79471031          0.68363      0.00000
## 195     694.1668   51  0.96513770          0.69845      0.00150
## 196     694.1668   45  0.45385554          0.71723      0.51100
## 197     694.1668   57  1.47641985          0.69689      0.00367
## 198     694.1668   31 -0.73913616          0.69576      0.01000
## 199     694.1668   14 -2.18776894          0.68112      0.01000
## 200     694.1668   67  2.32855678          0.67764      0.00600
## 201     694.1668   46  0.53906923          0.70639      0.05000
## 202     694.1668   59  1.64684724          0.72874      0.00000
## 203     694.1668   39 -0.05742662          0.69554      0.00247
## 204     694.1668   50  0.87992400          0.68269      0.00375
## 205     694.1668   55  1.30599247          0.71591      0.00278
## 206     694.1668   67  2.32855678          0.70510      0.00582
## 207     694.1668   62  1.90248832          0.66120      0.02300
## 208     694.1668   69  2.49898417          0.67580      0.03065
## 209     694.1668   48  0.70949662          0.65880      0.02200
## 210     694.1668   63  1.98770201          0.67696      0.00006
## 211     694.1668   51  0.96513770          0.67067      0.00000
## 212     694.1668   70  2.58419786          0.69107      0.00000
## 213     694.1668   57  1.47641985          0.68250      0.00000
## 214     694.1668   41  0.11300077          0.66049      0.03376
## 215     694.1668   35 -0.39828139          0.70989      0.00000
## 216     694.1668   38 -0.14264031          0.68650      0.02000
## 217     694.1668   55  1.30599247          0.70823      0.07544
## 218     694.1668   11 -2.44341002          0.71083      0.00020
## 219     694.1668   52  1.05035139          0.67604      0.10561
## 220     694.1668   57  1.47641985          0.71069      0.00400
## 221     694.1668   42  0.19821446          0.67492      0.11177
## 222     694.1668   84  3.77718957          0.70531      0.00639
## 223     694.1668   56  1.39120616          0.68030      0.00000
## 224     694.1668   50  0.87992400          0.68942      0.00000
## 225     694.1668   39 -0.05742662          0.70854      0.02028
## 226     694.1668   51  0.96513770          0.70502      0.00000
## 227     694.1668   49  0.79471031          0.72390      0.01065
## 228     694.1668   46  0.53906923          0.72209      0.00000
## 229     694.1668   50  0.87992400          0.73063      0.00150
## 230     694.1668   53  1.13556508          0.72127      0.00540
## 231     694.1668   59  1.64684724          0.71605      0.15000
## 232     694.1668   58  1.56163355          0.66143      0.00000
## 233     694.1668   70  2.58419786          0.67300      0.00006
## 234     694.1668   49  0.79471031          0.66321      0.03257
## 235     694.1668   48  0.70949662          0.67567      0.00500
## 236     694.1668   43  0.28342815          0.68516      0.00000
## 237     694.1668   44  0.36864185          0.67892      0.00000
## 238     694.1668   60  1.73206093          0.72470      0.00000
```

```
## 239     694.1668   51  0.96513770          0.70878      0.00600
## 240     694.1668   69  2.49898417          0.69373      0.00000
## 241     694.1668   45  0.45385554          0.67145      0.00007
## 242     694.1668   39 -0.05742662          0.67774      0.00000
## 243     694.1668   37 -0.22785401          0.71780      0.00678
## 244     694.1668   36 -0.31306770          0.65942      0.00300
## 245     694.1668   65  2.15812940          0.72935      0.00600
## 246     694.1668   47  0.62428292          0.70824      0.01187
## 247     694.1668   51  0.96513770          0.66158      0.06590
## 248     694.1668   60  1.73206093          0.66830      0.00168
## 249     694.1668   39 -0.05742662          0.66980      0.01210
## 250     694.1668   49  0.79471031          0.66628      0.00000
## 251     694.1668   54  1.22077878          0.66464      0.01000
## 252     694.1668   42  0.19821446          0.71774      0.00742
## 253     694.1668   29 -0.90956355          0.65963      0.00000
## 254     694.1668   70  2.58419786          0.66911      0.00000
## 255     694.1668   74  2.92505264          0.66584      0.00500
## 256     694.1668   44  0.36864185          0.67873      0.00000
## 257     694.1668   53  1.13556508          0.67526      0.00000
## 258     694.1668   53  1.13556508          0.70105      0.00000
## 259     694.1668   73  2.83983894          0.68252      0.00000
## 260     694.1668   40  0.02778707          0.69452      0.00000
## 261     694.1668   42  0.19821446          0.69452      0.00000
## 262     694.1668   51  0.96513770          0.72046      0.00000
## 263     694.1668   52  1.05035139          0.71579      0.00000
## 264     694.1668   27 -1.07999094          0.68433      0.07059
## 265     694.1668   37 -0.22785401          0.68748      0.01704
## 266     694.1668   23 -1.42084571          0.67958      0.00857
## 267     694.1668   32 -0.65392247          0.71972      0.00500
## 268     694.1668   19 -1.76170048          0.70622      0.06700
## 269     694.1668   40  0.02778707          0.68179      0.02503
## 270     694.1668   33 -0.56870878          0.66640      0.00000
## 271     694.1668   25 -1.25041832          0.71498      0.09800
## 272     694.1668   31 -0.73913616          0.71465      0.00196
## 273     694.1668   31 -0.73913616          0.68109      0.00000
## 274     694.1668   37 -0.22785401          0.69873      0.00014
## 275     694.1668   31 -0.73913616          0.68564      0.26726
## 276     694.1668   29 -0.90956355          0.73042      0.18027
## 277     694.1668   31 -0.73913616          0.70181      0.02000
## 278     694.1668   29 -0.90956355          0.67088      0.00099
## 279     694.1668   39 -0.05742662          0.71238      0.00444
## 280     694.1668   28 -0.99477724          0.69748      0.00000
## 281     694.1668   26 -1.16520463          0.71859      0.00000
## 282     694.1668   24 -1.33563201          0.68985      0.03000
## 283     694.1668   32 -0.65392247          0.70311      0.02010
## 284     694.1668   23 -1.42084571          0.66875      0.00013
## 285     694.1668   25 -1.25041832          0.70367      0.00000
## 286     694.1668   33 -0.56870878          0.70515      0.00000
## 287     694.1668   25 -1.25041832          0.72494      0.00722
## 288     694.1668   31 -0.73913616          0.69730      0.00000
## 289     694.1668   34 -0.48349508          0.72642      0.00000
## 290     694.1668   28 -0.99477724          0.68955      0.00000
## 291     694.1668   35 -0.39828139          0.72524      0.00040
## 292     694.1668   32 -0.65392247          0.72143      0.00000
```

```
## 293     694.1668   27 -1.07999094          0.70060      0.00000
## 294     694.1668   31 -0.73913616          0.68172      0.00000
## 295     694.1668   22 -1.50605940          0.70970      0.00000
## 296     694.1668   31 -0.73913616          0.72584      0.00000
## 297     694.1668   32 -0.65392247          0.70889      0.00000
## 298     694.1668   32 -0.65392247          0.70304      0.00204
## 299     694.1668   27 -1.07999094          0.72329      0.00001
## 300     694.1668   31 -0.73913616          0.70951      0.00000
## 301     694.1668   30 -0.82434986          0.67782      0.00098
## 302     694.1668   26 -1.16520463          0.70196      0.05552
## 303     694.1668   36 -0.31306770          0.70652      0.00105
## 304     694.1668   28 -0.99477724          0.71092      0.00001
## 305     694.1668   27 -1.07999094          0.66212      0.00000
## 306     694.1668   38 -0.14264031          0.70488      0.00000
## 307     694.1668   27 -1.07999094          0.67172      0.00001
## 308     694.1668   20 -1.67648679          0.68370      0.00444
## 309     694.1668   20 -1.67648679          0.68419      0.00006
## 310     694.1668   30 -0.82434986          0.71539      0.00000
## 311     694.1668   41  0.11300077          0.70703      0.00010
## 312     694.1668   24 -1.33563201          0.67567      0.00000
## 313     694.1668   23 -1.42084571          0.70199      0.00001
## 314     694.1668   26 -1.16520463          0.70141      0.00000
## 315     694.1668   26 -1.16520463          0.68022      0.00000
## 316     694.1668   35 -0.39828139          0.68848      0.00001
## 317     694.1668   27 -1.07999094          0.71649      0.00042
## 318     694.1668   33 -0.56870878          0.68457      0.00000
## 319     694.1668   32 -0.65392247          0.72666      0.00000
## 320     694.1668   30 -0.82434986          0.68262      0.00001
## 321     694.1668   31 -0.73913616          0.69065      0.00113
## 322     694.1668   35 -0.39828139          0.69002      0.00044
## 323     694.1668   32 -0.65392247          0.70616      0.02732
## 324     694.1668   20 -1.67648679          0.68336      0.00013
## 325     694.1668   23 -1.42084571          0.73182      0.00179
## 326     694.1668   26 -1.16520463          0.71083      0.00016
## 327     694.1668   31 -0.73913616          0.71454      0.00228
## 328     694.1668   38 -0.14264031          0.70162      0.00000
## 329     694.1668   30 -0.82434986          0.72138      0.00067
## 330     694.1668   24 -1.33563201          0.72567      0.00000
## 331     694.1668   24 -1.33563201          0.66749      0.00100
## 332     694.1668   35 -0.39828139          0.70774      0.00000
## 333     694.1668   32 -0.65392247          0.67047      0.00002
## 334     694.1668   27 -1.07999094          0.66262      0.00122
## 335     694.1668   26 -1.16520463          0.67613      0.00003
## 336     694.1668   32 -0.65392247          0.66618      0.00000
## 337     694.1668   38 -0.14264031          0.71897      0.00000
## 338     694.1668   33 -0.56870878          0.69492      0.00000
## 339     694.1668   31 -0.73913616          0.67590      0.00000
## 340     694.1668   23 -1.42084571          0.69202      0.00000
## 341     694.1668   33 -0.56870878          0.68786      0.04208
## 342     694.1668   39 -0.05742662          0.72353      0.00000
## 343     694.1668   32 -0.65392247          0.66797      0.00112
## 344     694.1668   29 -0.90956355          0.65814      0.01000
## 345     694.1668   25 -1.25041832          0.69454      0.00000
## 346     694.1668   30 -0.82434986          0.70641      0.00002
```

```
## 347     694.1668   29 -0.90956355          0.71053      0.00000
## 348     694.1668   33 -0.56870878          0.68331      0.00000
## 349     694.1668   28 -0.99477724          0.69434      0.00000
## 350     694.1668   29 -0.90956355          0.70112      0.01004
## 351     694.1668   34 -0.48349508          0.71898      0.00000
## 352     694.1668   30 -0.82434986          0.67455      0.00286
## 353     694.1668   31 -0.73913616          0.69987      0.04500
## 354     694.1668   25 -1.25041832          0.68444      0.00000
## 355     694.1668   22 -1.50605940          0.69088      0.00004
## 356     694.1668   32 -0.65392247          0.67079      0.01212
## 357     694.1668   19 -1.76170048          0.71483      0.00000
## 358     694.1668   29 -0.90956355          0.71158      0.00067
## 359     694.1668   32 -0.65392247          0.71004      0.00000
## 360     694.1668   40  0.02778707          0.71800      0.00000
## 361     694.1668   32 -0.65392247          0.70819      0.00000
## 362     694.1668   28 -0.99477724          0.73030      0.00000
## 363     694.1668   32 -0.65392247          0.70653      0.00000
## 364     694.1668   34 -0.48349508          0.67350      0.00001
## 365     694.1668   24 -1.33563201          0.73009      0.00195
## 366     694.1668   26 -1.16520463          0.70050      0.00001
## 367     694.1668   24 -1.33563201          0.69735      0.00017
## 368     694.1668   31 -0.73913616          0.70779      0.00000
## 369     694.1668   32 -0.65392247          0.68219      0.00140
## 370     694.1668   23 -1.42084571          0.69253      0.00000
## 371     694.1668   29 -0.90956355          0.68231      0.00018
## 372     694.1668   36 -0.31306770          0.68533      0.00006
## 373     694.1668   26 -1.16520463          0.71023      0.00146
## 374     694.1668   33 -0.56870878          0.70724      0.00022
## 375     694.1668   30 -0.82434986          0.68501      0.00055
## 376     694.1668   26 -1.16520463          0.66175      0.00000
## 377     694.1668   32 -0.65392247          0.70005      0.00070
## 378     694.1668   25 -1.25041832          0.68094      0.00500
## 379     694.1668   27 -1.07999094          0.68805      0.00060
## 380     694.1668   42  0.19821446          0.72657      0.00000
## 381     694.1668   20 -1.67648679          0.68979      0.00000
## 382     694.1668   33 -0.56870878          0.69928      0.00050
## 383     694.1668   29 -0.90956355          0.66680      0.00005
## 384     694.1668   37 -0.22785401          0.72109      0.00003
## 385     694.1668   40  0.02778707          0.65959      0.00000
## 386     694.1668   34 -0.48349508          0.71574      0.01058
## 387     694.1668   29 -0.90956355          0.68687      0.00000
## 388     694.1668   25 -1.25041832          0.72187      0.00000
## 389     694.1668   28 -0.99477724          0.70745      0.00001
## 390     694.1668   35 -0.39828139          0.70623      0.00027
## 391     694.1668   31 -0.73913616          0.67795      0.00300
## 392     694.1668   28 -0.99477724          0.70014      0.00003
## 393     694.1668   28 -0.99477724          0.73150      0.00000
## 394     694.1668   24 -1.33563201          0.72737      0.00001
## 395     694.1668   29 -0.90956355          0.69558      0.00019
## 396     694.1668   30 -0.82434986          0.69734      0.00000
## 397     694.1668   21 -1.59127309          0.72121      0.00316
## 398     694.1668   26 -1.16520463          0.67822      0.00050
## 399     694.1668   31 -0.73913616          0.66062      0.00350
## 400     694.1668   23 -1.42084571          0.69381      0.03552
```

```
## 401     694.1668   27 -1.07999094          0.68082      0.00000
## 402     694.1668   30 -0.82434986          0.68091      0.00687
## 403     694.1668   31 -0.73913616          0.67265      0.00000
## 404     694.1668   34 -0.48349508          0.69381      0.01950
## 405     694.1668   40  0.02778707          0.68418      0.00000
## 406     694.1668   47  0.62428292          0.71920      0.00053
## 407     694.1668   44  0.36864185          0.66636      0.00041
## 408     694.1668   26 -1.16520463          0.66074      0.00033
## 409     694.1668   42  0.19821446          0.68858      0.00000
## 410     694.1668   32 -0.65392247          0.71442      0.00100
## 411     694.1668   40  0.02778707          0.70247      0.01000
## 412     694.1668   39 -0.05742662          0.66350      0.00000
## 413     694.1668   26 -1.16520463          0.71585      0.00005
## 414     694.1668   59  1.64684724          0.72097      0.00000
## 415     694.1668   40  0.02778707          0.66902      0.00035
## 416     694.1668   54  1.22077878          0.67405      0.00080
## 417     694.1668   51  0.96513770          0.66382      0.00010
## 418     694.1668   48  0.70949662          0.70900      0.00006
## 419     694.1668   48  0.70949662          0.67572      0.00000
## 420     694.1668   36 -0.31306770          0.71302      0.02100
## 421     694.1668   44  0.36864185          0.71190      0.00001
## 422     694.1668   44  0.36864185          0.73101      0.02000
## 423     694.1668   23 -1.42084571          0.68337      0.04607
## 424     694.1668   37 -0.22785401          0.69196      0.02206
## 425     694.1668   52  1.05035139          0.72950      0.05860
## 426     694.1668   48  0.70949662          0.67805      0.01391
## 427     694.1668   40  0.02778707          0.70182      0.02279
## 428     694.1668   49  0.79471031          0.69104      0.00000
## 429     694.1668   24 -1.33563201          0.72614      0.00100
## 430     694.1668   30 -0.82434986          0.70781      0.00007
## 431     694.1668   20 -1.67648679          0.70895      0.00000
## 432     694.1668   55  1.30599247          0.70018      0.05000
## 433     694.1668   29 -0.90956355          0.70394      0.00591
## 434     694.1668   30 -0.82434986          0.68617      0.00500
## 435     694.1668   37 -0.22785401          0.66563      0.00019
## 436     694.1668   33 -0.56870878          0.71644      0.00300
## 437     694.1668   45  0.45385554          0.71946      0.00000
## 438     694.1668   44  0.36864185          0.69124      0.00000
## 439     694.1668   38 -0.14264031          0.69270      0.00000
## 440     694.1668   41  0.11300077          0.67382      0.00000
## 441     694.1668   30 -0.82434986          0.69406      0.00690
## 442     694.1668   22 -1.50605940          0.66423      0.00234
## 443     694.1668   35 -0.39828139          0.68073      0.00500
## 444     694.1668   33 -0.56870878          0.67373      0.00000
## 445     694.1668   42  0.19821446          0.69372      0.00000
## 446     694.1668   31 -0.73913616          0.67369      0.00000
## 447     694.1668   50  0.87992400          0.70696      0.02197
## 448     694.1668   42  0.19821446          0.72376      0.00050
## 449     694.1668   32 -0.65392247          0.69458      0.00090
## 450     694.1668   50  0.87992400          0.67741      0.00000
## 451     694.1668   52  1.05035139          0.68343      0.00000
## 452     694.1668   34 -0.48349508          0.70181      0.00000
## 453     694.1668   34 -0.48349508          0.68990      0.00000
## 454     694.1668   49  0.79471031          0.72890      0.00000
```

```
## 455     694.1668   45  0.45385554          0.67359      0.00340
## 456     694.1668   37 -0.22785401          0.69864      0.00020
## 457     694.1668   36 -0.31306770          0.71731      0.00000
## 458     694.1668   37 -0.22785401          0.70328      0.00000
## 459     694.1668   36 -0.31306770          0.70605      0.00347
## 460     694.1668   40  0.02778707          0.70523      0.00327
## 461     694.1668   35 -0.39828139          0.65998      0.04071
## 462     694.1668   39 -0.05742662          0.71379      0.01249
## 463     694.1668   42  0.19821446          0.67044      0.01344
## 464     694.1668   39 -0.05742662          0.67607      0.00948
## 465     694.1668   37 -0.22785401          0.69569      0.02079
## 466     694.1668   40  0.02778707          0.70303      0.03000
## 467     694.1668   55  1.30599247          0.71169      0.02384
## 468     694.1668   37 -0.22785401          0.70879      0.00200
## 469     694.1668   43  0.28342815          0.72191      0.00000
## 470     694.1668   30 -0.82434986          0.72757      0.00000
## 471     694.1668   54  1.22077878          0.69862      0.00000
## 472     694.1668   42  0.19821446          0.69554      0.00162
## 473     694.1668   31 -0.73913616          0.71705      0.01583
## 474     694.1668   42  0.19821446          0.73106      0.00058
## 475     694.1668   44  0.36864185          0.67154      0.00059
## 476     694.1668   41  0.11300077          0.66673      0.00540
## 477     694.1668   38 -0.14264031          0.69626      0.00000
## 478     694.1668   31 -0.73913616          0.70081      0.03290
## 479     694.1668   55  1.30599247          0.72034      0.00000
## 480     694.1668   45  0.45385554          0.70643      0.13212
## 481     694.1668   39 -0.05742662          0.70005      0.01000
## 482     694.1668   43  0.28342815          0.72616      0.00000
## 483     694.1668   51  0.96513770          0.67121      0.05777
## 484     694.1668   40  0.02778707          0.68181      0.01200
## 485     694.1668   39 -0.05742662          0.66155      0.00150
## 486     694.1668   47  0.62428292          0.68189      0.00000
## 487     694.1668   32 -0.65392247          0.72115      0.00006
## 488     694.1668   48  0.70949662          0.70370      0.00000
## 489     694.1668   50  0.87992400          0.68449      0.04159
## 490     694.1668   38 -0.14264031          0.68840      0.00918
## 491     694.1668   49  0.79471031          0.71997      0.00768
## 492     694.1668   33 -0.56870878          0.69871      0.00500
## 493     694.1668   33 -0.56870878          0.66073      0.00000
## 494     694.1668   31 -0.73913616          0.70371      0.00000
## 495     694.1668   34 -0.48349508          0.65895      0.00000
## 496     694.1668   37 -0.22785401          0.67027      0.00000
## 497     694.1668   40  0.02778707          0.68563      0.00000
## 498     694.1668   41  0.11300077          0.70457      0.00004
## 499     694.1668   50  0.87992400          0.67163      0.00001
## 500     694.1668   45  0.45385554          0.69861      0.00283
## 501     694.1668   48  0.70949662          0.66124      0.00007
## 502     694.1668   36 -0.31306770          0.69653      0.00009
## 503     694.1668   39 -0.05742662          0.66730      0.00375
## 504     694.1668   29 -0.90956355          0.70348      0.00010
## 505     694.1668   26 -1.16520463          0.66274      0.00000
## 506     694.1668   36 -0.31306770          0.70316      0.00000
## 507     694.1668   35 -0.39828139          0.70450      0.00001
## 508     694.1668   51  0.96513770          0.72067      0.00000
```

```
## 509     694.1668   50  0.87992400          0.66983      0.00000
## 510     694.1668   51  0.96513770          0.66245      0.01000
## 511     694.1668   22 -1.50605940          0.67465      0.00809
## 512     694.1668   43  0.28342815          0.71251      0.00000
## 513     694.1668   34 -0.48349508          0.69065      0.00000
## 514     694.1668   33 -0.56870878          0.68226      0.00004
## 515     694.1668   41  0.11300077          0.72109      0.00587
## 516     694.1668   38 -0.14264031          0.66245      0.00350
## 517     694.1668   54  1.22077878          0.68178      0.00000
## 518     694.1668   50  0.87992400          0.69187      0.00000
## 519     694.1668   40  0.02778707          0.73089      0.00002
## 520     694.1668   35 -0.39828139          0.66292      0.00501
## 521     694.1668   43  0.28342815          0.67200      0.00351
## 522     694.1668   36 -0.31306770          0.70198      0.00160
## 523     694.1668   38 -0.14264031          0.72829      0.00000
## 524     694.1668   44  0.36864185          0.73085      0.00011
## 525     694.1668   45  0.45385554          0.67635      0.00021
## 526     694.1668   36 -0.31306770          0.72155      0.00390
## 527     694.1668   41  0.11300077          0.66767      0.00018
## 528     694.1668   35 -0.39828139          0.67712      0.00154
## 529     694.1668   50  0.87992400          0.66710      0.00000
## 530     694.1668   47  0.62428292          0.71345      0.00000
## 531     694.1668   45  0.45385554          0.67532      0.00005
## 532     694.1668   37 -0.22785401          0.68054      0.00000
## 533     694.1668   38 -0.14264031          0.70614      0.00117
## 534     694.1668   38 -0.14264031          0.66098      0.00017
## 535     694.1668   34 -0.48349508          0.72996      0.00246
## 536     694.1668   41  0.11300077          0.72224      0.00237
## 537     694.1668   35 -0.39828139          0.70205      0.00001
## 538     694.1668   44  0.36864185          0.68977      0.00364
## 539     694.1668   35 -0.39828139          0.68287      0.00003
## 540     694.1668   46  0.53906923          0.66536      0.00001
## 541     694.1668   42  0.19821446          0.71615      0.00750
## 542     753.3742   62  1.90248832          0.74170      0.00000
## 543     753.3742   54  1.22077878          0.73369      0.00000
## 544     753.3742   52  1.05035139          0.73613      0.00000
## 545     753.3742   46  0.53906923          0.78731      0.10000
## 546     753.3742   70  2.58419786          0.77117      0.00000
## 547     753.3742   35 -0.39828139          0.75383      0.00560
## 548     753.3742   45  0.45385554          0.73776      0.00300
## 549     753.3742   47  0.62428292          0.74368      0.00200
## 550     753.3742   44  0.36864185          0.73852      0.00000
## 551     753.3742   40  0.02778707          0.73688      0.00000
## 552     753.3742   48  0.70949662          0.76669      0.00000
## 553     753.3742   65  2.15812940          0.76537      0.00730
## 554     753.3742   54  1.22077878          0.73970      0.00000
## 555     753.3742   56  1.39120616          0.74427      0.00500
## 556     753.3742   57  1.47641985          0.78651      0.00000
## 557     753.3742   53  1.13556508          0.78281      0.00020
## 558     753.3742   44  0.36864185          0.75496      0.00000
## 559     753.3742   38 -0.14264031          0.75221      0.00000
## 560     753.3742   48  0.70949662          0.78052      0.00000
## 561     753.3742   49  0.79471031          0.74209      0.00000
## 562     753.3742   55  1.30599247          0.75267      0.00000
```

```
## 563     753.3742   46  0.53906923          0.73833      0.00000
## 564     753.3742   54  1.22077878          0.75372      0.00225
## 565     753.3742   52  1.05035139          0.75646      0.00000
## 566     753.3742   63  1.98770201          0.73847      0.00001
## 567     753.3742   53  1.13556508          0.74420      0.00000
## 568     753.3742   61  1.81727463          0.73878      0.00000
## 569     753.3742   51  0.96513770          0.73308      0.00000
## 570     753.3742   55  1.30599247          0.74171      0.00147
## 571     753.3742   51  0.96513770          0.75245      0.00000
## 572     753.3742   41  0.11300077          0.75425      0.00000
## 573     753.3742   52  1.05035139          0.73615      0.00050
## 574     753.3742   51  0.96513770          0.76167      0.01000
## 575     753.3742   54  1.22077878          0.74073      0.02510
## 576     753.3742   74  2.92505264          0.74852      0.00809
## 577     753.3742   55  1.30599247          0.75649      0.00000
## 578     753.3742   57  1.47641985          0.75931      0.00000
## 579     753.3742   54  1.22077878          0.73656      0.00827
## 580     753.3742   53  1.13556508          0.73279      0.01602
## 581     753.3742   44  0.36864185          0.73564      0.00000
## 582     753.3742   46  0.53906923          0.74186      0.00000
## 583     753.3742   48  0.70949662          0.78278      0.00837
## 584     753.3742   60  1.73206093          0.74520      0.00000
## 585     753.3742   50  0.87992400          0.79511      0.00000
## 586     753.3742   46  0.53906923          0.74063      0.00158
## 587     753.3742   54  1.22077878          0.76406      0.03000
## 588     753.3742   33 -0.56870878          0.75755      0.01655
## 589     753.3742   27 -1.07999094          0.74140      0.00000
## 590     753.3742   39 -0.05742662          0.77216      0.00790
## 591     753.3742   53  1.13556508          0.80020      0.00396
## 592     753.3742   50  0.87992400          0.80580      0.00000
## 593     753.3742   34 -0.48349508          0.74419      0.00000
## 594     753.3742   53  1.13556508          0.74260      0.00000
## 595     753.3742   42  0.19821446          0.74473      0.08851
## 596     753.3742   51  0.96513770          0.75494      0.00000
## 597     753.3742   27 -1.07999094          0.73268      0.00000
## 598     753.3742   22 -1.50605940          0.75147      0.00000
## 599     753.3742   35 -0.39828139          0.73272      0.00300
## 600     753.3742   28 -0.99477724          0.75773      0.01200
## 601     753.3742   33 -0.56870878          0.75644      0.01000
## 602     753.3742   22 -1.50605940          0.74976      0.00000
## 603     753.3742   32 -0.65392247          0.75691      0.00000
## 604     753.3742   29 -0.90956355          0.77781      0.00000
## 605     753.3742   26 -1.16520463          0.73734      0.00032
## 606     753.3742   33 -0.56870878          0.77861      0.00000
## 607     753.3742   27 -1.07999094          0.73729      0.00126
## 608     753.3742   25 -1.25041832          0.77590      0.00004
## 609     753.3742   39 -0.05742662          0.75231      0.00021
## 610     753.3742   17 -1.93212787          0.75069      0.00000
## 611     753.3742   30 -0.82434986          0.75691      0.00025
## 612     753.3742   22 -1.50605940          0.76474      0.00000
## 613     753.3742   25 -1.25041832          0.78593      0.00020
## 614     753.3742   34 -0.48349508          0.73219      0.00204
## 615     753.3742   32 -0.65392247          0.76111      0.00008
## 616     753.3742   31 -0.73913616          0.74860      0.00000
```

```
## 617     753.3742   24 -1.33563201          0.75294      0.00000
## 618     753.3742   33 -0.56870878          0.76785      0.00000
## 619     753.3742   28 -0.99477724          0.74912      0.04000
## 620     753.3742   29 -0.90956355          0.73251      0.00012
## 621     753.3742   35 -0.39828139          0.74006      0.04000
## 622     753.3742   25 -1.25041832          0.73203      0.00001
## 623     753.3742   29 -0.90956355          0.76748      0.00000
## 624     753.3742   32 -0.65392247          0.75237      0.01300
## 625     753.3742   27 -1.07999094          0.75943      0.00016
## 626     753.3742   28 -0.99477724          0.74952      0.00000
## 627     753.3742   36 -0.31306770          0.73355      0.00000
## 628     753.3742   32 -0.65392247          0.76179      0.00000
## 629     753.3742   32 -0.65392247          0.73525      0.00184
## 630     753.3742   36 -0.31306770          0.78835      0.00000
## 631     753.3742   30 -0.82434986          0.74279      0.00000
## 632     753.3742   35 -0.39828139          0.73232      0.00002
## 633     753.3742   22 -1.50605940          0.74992      0.00008
## 634     753.3742   34 -0.48349508          0.75895      0.00000
## 635     753.3742   24 -1.33563201          0.75904      0.00000
## 636     753.3742   31 -0.73913616          0.73238      0.00000
## 637     753.3742   28 -0.99477724          0.76283      0.00769
## 638     753.3742   29 -0.90956355          0.74125      0.00000
## 639     753.3742   32 -0.65392247          0.73835      0.00001
## 640     753.3742   25 -1.25041832          0.74729      0.00000
## 641     753.3742   32 -0.65392247          0.74825      0.05200
## 642     753.3742   32 -0.65392247          0.75404      0.00001
## 643     753.3742   23 -1.42084571          0.73934      0.00000
## 644     753.3742   24 -1.33563201          0.75407      0.00000
## 645     753.3742   31 -0.73913616          0.73212      0.00018
## 646     753.3742   36 -0.31306770          0.73820      0.00010
## 647     753.3742   32 -0.65392247          0.74666      0.00044
## 648     753.3742   44  0.36864185          0.77379      0.00000
## 649     753.3742   33 -0.56870878          0.77814      0.00351
## 650     753.3742   44  0.36864185          0.73705      0.00475
## 651     753.3742   46  0.53906923          0.74700      0.00000
## 652     753.3742   31 -0.73913616          0.79609      0.00000
## 653     753.3742   45  0.45385554          0.73332      0.00000
## 654     753.3742   46  0.53906923          0.75433      0.00028
## 655     753.3742   46  0.53906923          0.75899      0.00099
## 656     753.3742   42  0.19821446          0.77129      0.00000
## 657     753.3742   46  0.53906923          0.74243      0.00990
## 658     753.3742   32 -0.65392247          0.74974      0.00001
## 659     753.3742   30 -0.82434986          0.74940      0.00028
## 660     753.3742   36 -0.31306770          0.75756      0.00000
## 661     753.3742   37 -0.22785401          0.73792      0.00321
## 662     753.3742   50  0.87992400          0.74375      0.00000
## 663     753.3742   43  0.28342815          0.73863      0.00458
## 664     753.3742   55  1.30599247          0.77941      0.00001
## 665     753.3742   46  0.53906923          0.80600      0.01400
## 666     753.3742   42  0.19821446          0.77726      0.08000
## 667     753.3742   27 -1.07999094          0.73695      0.01097
## 668     753.3742   45  0.45385554          0.76674      0.00000
## 669     753.3742   28 -0.99477724          0.76091      0.00000
## 670     753.3742   49  0.79471031          0.75307      0.05124
```

```
## 671     753.3742   29 -0.90956355              0.74489      0.00001
## 672     753.3742   49  0.79471031              0.78779      0.00000
## 673     753.3742   28 -0.99477724              0.78898      0.00000
## 674     753.3742   43  0.28342815              0.73605      0.00000
## 675     753.3742   48  0.70949662              0.73686      0.00000
## 676     753.3742   39 -0.05742662              0.74658      0.00108
## 677     753.3742   33 -0.56870878              0.75832      0.00000
## 678     753.3742   32 -0.65392247              0.75429      0.00000
## 679     753.3742   34 -0.48349508              0.76384      0.00200
## 680     753.3742   29 -0.90956355              0.78506      0.00068
## 681     753.3742   45  0.45385554              0.75092      0.00000
## 682     753.3742   55  1.30599247              0.74015      0.00000
## 683     753.3742   45  0.45385554              0.74355      0.00000
## 684     753.3742   35 -0.39828139              0.75356      0.00150
## 685     753.3742   56  1.39120616              0.74084      0.00003
## 686     753.3742   50  0.87992400              0.74554      0.00000
## 687     753.3742   41  0.11300077              0.75888      0.00537
## 688     753.3742   42  0.19821446              0.75254      0.00002
## 689     753.3742   34 -0.48349508              0.73622      0.00000
## 690     753.3742   37 -0.22785401              0.73994      0.00000
##     credit.score_bins
## 1                 low
## 2                 low
## 3                 low
## 4                 low
## 5                 low
## 6                 low
## 7                 low
## 8                 low
## 9                 low
## 10                low
## 11                low
## 12                low
## 13                low
## 14                low
## 15                low
## 16                low
## 17                low
## 18                low
## 19                low
## 20                low
## 21                low
## 22                low
## 23                low
## 24                low
## 25                low
## 26                low
## 27                low
## 28                low
## 29                low
## 30                low
## 31                low
## 32                low
## 33                low
```

```
## 34              low
## 35              low
## 36              low
## 37              low
## 38              low
## 39              low
## 40              low
## 41              low
## 42              low
## 43              low
## 44              low
## 45              low
## 46              low
## 47              low
## 48              low
## 49              low
## 50              low
## 51              low
## 52              low
## 53              low
## 54              low
## 55              low
## 56              low
## 57              low
## 58              low
## 59              low
## 60              low
## 61              low
## 62              low
## 63              low
## 64              low
## 65              low
## 66              low
## 67              low
## 68              low
## 69              low
## 70              low
## 71              low
## 72              low
## 73              low
## 74              low
## 75              low
## 76              low
## 77              low
## 78              low
## 79              low
## 80              low
## 81              low
## 82              low
## 83              low
## 84              low
## 85              low
## 86              low
## 87              low
```

```
## 88              low
## 89              low
## 90              low
## 91              low
## 92              low
## 93              low
## 94              low
## 95              low
## 96              low
## 97              low
## 98              low
## 99              low
## 100             low
## 101             low
## 102             low
## 103             low
## 104             low
## 105             low
## 106             low
## 107             low
## 108             low
## 109             low
## 110             low
## 111             low
## 112             low
## 113             low
## 114             low
## 115             low
## 116             low
## 117             low
## 118             low
## 119             low
## 120             low
## 121             low
## 122             low
## 123             low
## 124          medium
## 125          medium
## 126          medium
## 127          medium
## 128          medium
## 129          medium
## 130          medium
## 131          medium
## 132          medium
## 133          medium
## 134          medium
## 135          medium
## 136          medium
## 137          medium
## 138          medium
## 139          medium
## 140          medium
## 141          medium
```

```
## 142            medium
## 143            medium
## 144            medium
## 145            medium
## 146            medium
## 147            medium
## 148            medium
## 149            medium
## 150            medium
## 151            medium
## 152            medium
## 153            medium
## 154            medium
## 155            medium
## 156            medium
## 157            medium
## 158            medium
## 159            medium
## 160            medium
## 161            medium
## 162            medium
## 163            medium
## 164            medium
## 165            medium
## 166            medium
## 167            medium
## 168            medium
## 169            medium
## 170            medium
## 171            medium
## 172            medium
## 173            medium
## 174            medium
## 175            medium
## 176            medium
## 177            medium
## 178            medium
## 179            medium
## 180            medium
## 181            medium
## 182            medium
## 183            medium
## 184            medium
## 185            medium
## 186            medium
## 187            medium
## 188            medium
## 189            medium
## 190            medium
## 191            medium
## 192            medium
## 193            medium
## 194            medium
## 195            medium
```

```
## 196            medium
## 197            medium
## 198            medium
## 199            medium
## 200            medium
## 201            medium
## 202            medium
## 203            medium
## 204            medium
## 205            medium
## 206            medium
## 207            medium
## 208            medium
## 209            medium
## 210            medium
## 211            medium
## 212            medium
## 213            medium
## 214            medium
## 215            medium
## 216            medium
## 217            medium
## 218            medium
## 219            medium
## 220            medium
## 221            medium
## 222            medium
## 223            medium
## 224            medium
## 225            medium
## 226            medium
## 227            medium
## 228            medium
## 229            medium
## 230            medium
## 231            medium
## 232            medium
## 233            medium
## 234            medium
## 235            medium
## 236            medium
## 237            medium
## 238            medium
## 239            medium
## 240            medium
## 241            medium
## 242            medium
## 243            medium
## 244            medium
## 245            medium
## 246            medium
## 247            medium
## 248            medium
## 249            medium
```

```
## 250          medium
## 251          medium
## 252          medium
## 253          medium
## 254          medium
## 255          medium
## 256          medium
## 257          medium
## 258          medium
## 259          medium
## 260          medium
## 261          medium
## 262          medium
## 263          medium
## 264          medium
## 265          medium
## 266          medium
## 267          medium
## 268          medium
## 269          medium
## 270          medium
## 271          medium
## 272          medium
## 273          medium
## 274          medium
## 275          medium
## 276          medium
## 277          medium
## 278          medium
## 279          medium
## 280          medium
## 281          medium
## 282          medium
## 283          medium
## 284          medium
## 285          medium
## 286          medium
## 287          medium
## 288          medium
## 289          medium
## 290          medium
## 291          medium
## 292          medium
## 293          medium
## 294          medium
## 295          medium
## 296          medium
## 297          medium
## 298          medium
## 299          medium
## 300          medium
## 301          medium
## 302          medium
## 303          medium
```

```
## 304            medium
## 305            medium
## 306            medium
## 307            medium
## 308            medium
## 309            medium
## 310            medium
## 311            medium
## 312            medium
## 313            medium
## 314            medium
## 315            medium
## 316            medium
## 317            medium
## 318            medium
## 319            medium
## 320            medium
## 321            medium
## 322            medium
## 323            medium
## 324            medium
## 325            medium
## 326            medium
## 327            medium
## 328            medium
## 329            medium
## 330            medium
## 331            medium
## 332            medium
## 333            medium
## 334            medium
## 335            medium
## 336            medium
## 337            medium
## 338            medium
## 339            medium
## 340            medium
## 341            medium
## 342            medium
## 343            medium
## 344            medium
## 345            medium
## 346            medium
## 347            medium
## 348            medium
## 349            medium
## 350            medium
## 351            medium
## 352            medium
## 353            medium
## 354            medium
## 355            medium
## 356            medium
## 357            medium
```

```
## 358          medium
## 359          medium
## 360          medium
## 361          medium
## 362          medium
## 363          medium
## 364          medium
## 365          medium
## 366          medium
## 367          medium
## 368          medium
## 369          medium
## 370          medium
## 371          medium
## 372          medium
## 373          medium
## 374          medium
## 375          medium
## 376          medium
## 377          medium
## 378          medium
## 379          medium
## 380          medium
## 381          medium
## 382          medium
## 383          medium
## 384          medium
## 385          medium
## 386          medium
## 387          medium
## 388          medium
## 389          medium
## 390          medium
## 391          medium
## 392          medium
## 393          medium
## 394          medium
## 395          medium
## 396          medium
## 397          medium
## 398          medium
## 399          medium
## 400          medium
## 401          medium
## 402          medium
## 403          medium
## 404          medium
## 405          medium
## 406          medium
## 407          medium
## 408          medium
## 409          medium
## 410          medium
## 411          medium
```

```
## 412            medium
## 413            medium
## 414            medium
## 415            medium
## 416            medium
## 417            medium
## 418            medium
## 419            medium
## 420            medium
## 421            medium
## 422            medium
## 423            medium
## 424            medium
## 425            medium
## 426            medium
## 427            medium
## 428            medium
## 429            medium
## 430            medium
## 431            medium
## 432            medium
## 433            medium
## 434            medium
## 435            medium
## 436            medium
## 437            medium
## 438            medium
## 439            medium
## 440            medium
## 441            medium
## 442            medium
## 443            medium
## 444            medium
## 445            medium
## 446            medium
## 447            medium
## 448            medium
## 449            medium
## 450            medium
## 451            medium
## 452            medium
## 453            medium
## 454            medium
## 455            medium
## 456            medium
## 457            medium
## 458            medium
## 459            medium
## 460            medium
## 461            medium
## 462            medium
## 463            medium
## 464            medium
## 465            medium
```

```
## 466            medium
## 467            medium
## 468            medium
## 469            medium
## 470            medium
## 471            medium
## 472            medium
## 473            medium
## 474            medium
## 475            medium
## 476            medium
## 477            medium
## 478            medium
## 479            medium
## 480            medium
## 481            medium
## 482            medium
## 483            medium
## 484            medium
## 485            medium
## 486            medium
## 487            medium
## 488            medium
## 489            medium
## 490            medium
## 491            medium
## 492            medium
## 493            medium
## 494            medium
## 495            medium
## 496            medium
## 497            medium
## 498            medium
## 499            medium
## 500            medium
## 501            medium
## 502            medium
## 503            medium
## 504            medium
## 505            medium
## 506            medium
## 507            medium
## 508            medium
## 509            medium
## 510            medium
## 511            medium
## 512            medium
## 513            medium
## 514            medium
## 515            medium
## 516            medium
## 517            medium
## 518            medium
## 519            medium
```

```
## 520          medium
## 521          medium
## 522          medium
## 523          medium
## 524          medium
## 525          medium
## 526          medium
## 527          medium
## 528          medium
## 529          medium
## 530          medium
## 531          medium
## 532          medium
## 533          medium
## 534          medium
## 535          medium
## 536          medium
## 537          medium
## 538          medium
## 539          medium
## 540          medium
## 541          medium
## 542            high
## 543            high
## 544            high
## 545            high
## 546            high
## 547            high
## 548            high
## 549            high
## 550            high
## 551            high
## 552            high
## 553            high
## 554            high
## 555            high
## 556            high
## 557            high
## 558            high
## 559            high
## 560            high
## 561            high
## 562            high
## 563            high
## 564            high
## 565            high
## 566            high
## 567            high
## 568            high
## 569            high
## 570            high
## 571            high
## 572            high
## 573            high
```

```
## 574              high
## 575              high
## 576              high
## 577              high
## 578              high
## 579              high
## 580              high
## 581              high
## 582              high
## 583              high
## 584              high
## 585              high
## 586              high
## 587              high
## 588              high
## 589              high
## 590              high
## 591              high
## 592              high
## 593              high
## 594              high
## 595              high
## 596              high
## 597              high
## 598              high
## 599              high
## 600              high
## 601              high
## 602              high
## 603              high
## 604              high
## 605              high
## 606              high
## 607              high
## 608              high
## 609              high
## 610              high
## 611              high
## 612              high
## 613              high
## 614              high
## 615              high
## 616              high
## 617              high
## 618              high
## 619              high
## 620              high
## 621              high
## 622              high
## 623              high
## 624              high
## 625              high
## 626              high
## 627              high
```

```
## 628              high
## 629              high
## 630              high
## 631              high
## 632              high
## 633              high
## 634              high
## 635              high
## 636              high
## 637              high
## 638              high
## 639              high
## 640              high
## 641              high
## 642              high
## 643              high
## 644              high
## 645              high
## 646              high
## 647              high
## 648              high
## 649              high
## 650              high
## 651              high
## 652              high
## 653              high
## 654              high
## 655              high
## 656              high
## 657              high
## 658              high
## 659              high
## 660              high
## 661              high
## 662              high
## 663              high
## 664              high
## 665              high
## 666              high
## 667              high
## 668              high
## 669              high
## 670              high
## 671              high
## 672              high
## 673              high
## 674              high
## 675              high
## 676              high
## 677              high
## 678              high
## 679              high
## 680              high
## 681              high
```

```
## 682              high
## 683              high
## 684              high
## 685              high
## 686              high
## 687              high
## 688              high
## 689              high
## 690              high
```

Problem 2

This is the first homework problem using machine learning algorithms. You will perform a straightforward training and evaluation of a support vector machine on the bank data from Problem 1. Start with a fresh copy, but be sure to remove rows with missing values first.

```
bank <-read.csv("/Users/kavanamanvi/Desktop/FDS/HW2/BankData.csv")
bank <- na.omit(bank)
summary(bank)
```

```
##        X             cont1           cont2            cont3
##  Min.   :  1.0   Min.   :13.75   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:172.2   1st Qu.:22.60   1st Qu.: 1.010   1st Qu.: 0.165
##  Median :347.5   Median :28.50   Median : 2.750   Median : 1.000
##  Mean   :345.8   Mean   :31.57   Mean   : 4.798   Mean   : 2.222
##  3rd Qu.:519.8   3rd Qu.:38.25   3rd Qu.: 7.207   3rd Qu.: 2.585
##  Max.   :690.0   Max.   :80.25   Max.   :28.000   Max.   :28.500
##     bool1              bool2              cont4           bool3
##  Length:666         Length:666         Min.   : 0.000   Length:666
##  Class :character   Class :character   1st Qu.: 0.000   Class :character
##  Mode  :character   Mode  :character   Median : 0.000   Mode  :character
##                                        Mean   : 2.459
##                                        3rd Qu.: 3.000
##                                        Max.   :67.000
##      cont5             cont6            approval          credit.score
##  Min.   :   0.00   Min.   :     0.0   Length:666         Min.   :585.1
##  1st Qu.:  75.25   1st Qu.:     0.0   Class :character   1st Qu.:666.4
##  Median : 160.00   Median :     5.0   Mode  :character   Median :697.1
##  Mean   : 182.12   Mean   :   998.6                      Mean   :696.3
##  3rd Qu.: 271.00   3rd Qu.:   399.0                      3rd Qu.:726.4
##  Max.   :2000.00   Max.   :100000.0                      Max.   :806.0
##       ages
##  Min.   :11.0
##  1st Qu.:31.0
##  Median :38.0
##  Mean   :39.7
##  3rd Qu.:48.0
##  Max.   :84.0
```

Pre-processing: since SVM only works with numerical data, convert boolean values to 0 and 1 instead of t and f.

```r
# Convert boolean values to 0 and 1
bank$bool1 <- as.integer(bank$bool1 == "t")
bank$bool2 <- as.integer(bank$bool2 == "t")
bank$bool3 <- as.integer(bank$bool3 == "t")

# Print the updated dataset
head(bank)
```

```
##   X cont1 cont2 cont3 bool1 bool2 cont4 bool3 cont5 cont6 approval credit.score
## 1 1 30.83 0.000  1.25     1     1     1     0   202     0        +       664.60
## 2 2 58.67 4.460  3.04     1     1     6     0    43   560        +       693.88
## 3 3 24.50 0.500  1.50     1     0     0     0   280   824        +       621.82
## 4 4 27.83 1.540  3.75     1     1     5     1   100     3        +       653.97
## 5 5 20.17 5.625  1.71     1     0     0     0   120     0        +       670.26
## 6 6 32.08 4.000  2.50     1     0     0     1   360     0        +       672.16
##   ages
## 1   42
## 2   54
## 3   29
## 4   58
## 5   65
## 6   61
```

a. Apply SVM to the data from Problem 1 to predict approval and report the accuracy using 10-fold cross validation.

```r
library(caret)
library(e1071)
train_control_cv = trainControl(method = "cv", number = 10)
svm_bank <- train(approval ~., data = bank, method = "svmLinear",
                  trControl = train_control_cv)
svm_bank
```

```
## Support Vector Machines with Linear Kernel
##
## 666 samples
##  12 predictor
##   2 classes: '-', '+'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 599, 599, 600, 600, 599, 599, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.863365  0.7288575
##
## Tuning parameter 'C' was held constant at a value of 1
```

b. Next, use the grid search functionality when training to optimize the C parameter of the SVM. What parameter was chosen and what is the accuracy?

The grid search functionality was used to optimize the C parameter of the SVM. The grid search tested a range of C values from 10^-5 to 10^2, with a step size of 0.5. The best C parameter chosen was 0.01, which resulted in an accuracy of approximately 86.35%. This means that the SVM model with a C parameter of 0.01 achieved the highest accuracy compared to other C values tested during the grid search.

```
grid <- expand.grid(C = 10^seq(-5,2,0.5))
svm_grid <- train(approval ~., data = bank, method = "svmLinear",
                  trControl = train_control_cv, tuneGrid = grid)
svm_grid
```

```
## Support Vector Machines with Linear Kernel
##
## 666 samples
##  12 predictor
##   2 classes: '-', '+'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 600, 599, 599, 600, 599, 599, ...
## Resampling results across tuning parameters:
##
##   C             Accuracy   Kappa
##   1.000000e-05  0.5510427  0.0000000
##   3.162278e-05  0.5510427  0.0000000
##   1.000000e-04  0.5510427  0.0000000
##   3.162278e-04  0.5705587  0.0472890
##   1.000000e-03  0.8378944  0.6706027
##   3.162278e-03  0.8662746  0.7344306
##   1.000000e-02  0.8632895  0.7284717
##   3.162278e-02  0.8632895  0.7284717
##   1.000000e-01  0.8632895  0.7284717
##   3.162278e-01  0.8632895  0.7284717
##   1.000000e+00  0.8632895  0.7284717
##   3.162278e+00  0.8632895  0.7284717
##   1.000000e+01  0.8632895  0.7284717
##   3.162278e+01  0.8632895  0.7284717
##   1.000000e+02  0.8632895  0.7284717
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 0.003162278.
```

c. Sometimes even if the grid of parameters in (b) includes the default value of C = 1 (used in (a)), the accuracy result will be different for this value of C. What could make that different?

The grid search in (a) explored a range of C values from 10^-5 to 10^2, with a step size of 0.5, and found that the optimal C value was 0.01, resulting in an accuracy of approximately 86.35%.

In contrast, the grid search in (b) used a fixed value of C = 1 and achieved an accuracy of approximately 86.33%.

The difference in accuracy between the two cases could be due to the nature of the dataset and the performance characteristics of the SVM algorithm. Grid search helps find the optimal C value by evaluating the model's performance across different values of C. Even though C = 1 is the default value, it may not always be the best choice for a given dataset. Factors such as the complexity of the dataset, the presence of outliers,

and the distribution of the classes can all influence the performance of the SVM model with different values of C. As a result, the grid search may find a different optimal C value that better suits the characteristics of the dataset, leading to a different accuracy result compared to using the default value of C = 1.

```
grid <- expand.grid(C = 1)
svm_grid <- train(approval ~., data = bank, method = "svmLinear",
                  trControl = train_control_cv, tuneGrid = grid)
svm_grid
```

```
## Support Vector Machines with Linear Kernel
##
## 666 samples
##  12 predictor
##   2 classes: '-', '+'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 599, 599, 599, 599, 599, 599, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8633347  0.728446
##
## Tuning parameter 'C' was held constant at a value of 1
```

Problem 3

We will take SVM further in this problem, showing how it often gets used even when the data are not suitable, by first engineering the numerical features we need. There is a Star Wars dataset in the dplyr library. Load that library and you will be able to see it (head(starwars)). There are some variables we will not use, so first remove films, vehicles, starships and name. Also remove rows with missing values.

```
library(dplyr)
data(starwars)
starwars <- starwars %>% select(-c("films", "vehicles", "starships", "name"))
names(starwars)
```

```
##  [1] "height"     "mass"       "hair_color" "skin_color" "eye_color"
##  [6] "birth_year" "sex"        "gender"     "homeworld"  "species"
```

```
#remove missing coloumns
starwars<- na.omit(starwars)
summary(starwars)
```

```
##      height           mass         hair_color         skin_color
##  Min.   : 88.0   Min.   : 20.00   Length:29          Length:29
##  1st Qu.:172.0   1st Qu.: 75.00   Class :character   Class :character
##  Median :180.0   Median : 79.00   Mode  :character   Mode  :character
##  Mean   :178.7   Mean   : 77.77
##  3rd Qu.:188.0   3rd Qu.: 83.00
##  Max.   :228.0   Max.   :136.00
```

```
##   eye_color           birth_year         sex                gender
##  Length:29          Min.   :  8.00   Length:29          Length:29
##  Class :character   1st Qu.: 31.00   Class :character   Class :character
##  Mode  :character   Median : 46.00   Mode  :character   Mode  :character
##                     Mean   : 51.29
##                     3rd Qu.: 57.00
##                     Max.   :200.00
##   homeworld           species
##  Length:29          Length:29
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

a. Several variables are categorical. We will use dummy variables to make it possible for SVM to use these. Leave the gender category out of the dummy variable conversion to use as a categorical for prediction. Show the resulting head. Convert categorical variables to dummy variables, leaving out 'gender'

```r
library(caret)

# Selecting only the categorical variables
categorical_vars <- c("hair_color", "skin_color", "eye_color", "homeworld", "species","sex")

# Creating dummy variables, excluding 'gender'
dummy <- dummyVars(~ ., data = starwars[, categorical_vars], levelsOnly = TRUE)

# Using the dummy predictor to transform the dataset
dummies <- as.data.frame(predict(dummy, newdata = starwars[, categorical_vars]))

# Combining the dummy variables with the original dataset
starwars_dummies <- cbind(starwars, dummies)

# Removing the original categorical variables
starwars_dummies <- starwars_dummies[, -which(names(starwars_dummies) %in% categorical_vars)]

head(starwars_dummies)
```

```
##   height mass birth_year    gender hair_colorauburn, white hair_colorblack
## 1    172   77       19.0 masculine                       0               0
## 2    202  136       41.9 masculine                       0               0
## 3    150   49       19.0  feminine                       0               0
## 4    178  120       52.0 masculine                       0               0
## 5    165   75       47.0  feminine                       0               0
## 6    183   84       24.0 masculine                       0               1
##   hair_colorblond hair_colorbrown hair_colorbrown, grey hair_colorgrey
## 1               1               0                     0              0
## 2               0               0                     0              0
## 3               0               1                     0              0
## 4               0               0                     1              0
## 5               0               1                     0              0
## 6               0               0                     0              0
```

```
##   hair_colornone hair_colorwhite skin_colorblue skin_colorbrown
## 1              0               0              0               0
## 2              1               0              0               0
## 3              0               0              0               0
## 4              0               0              0               0
## 5              0               0              0               0
## 6              0               0              0               0
##   skin_colorbrown mottle skin_colordark skin_colorfair skin_colorgreen
## 1                       0              0              1               0
## 2                       0              0              0               0
## 3                       0              0              0               0
## 4                       0              0              0               0
## 5                       0              0              0               0
## 6                       0              0              0               0
##   skin_colorlight skin_colororange skin_colorpale skin_colorred skin_colortan
## 1               0                0              0             0             0
## 2               0                0              0             0             0
## 3               1                0              0             0             0
## 4               1                0              0             0             0
## 5               1                0              0             0             0
## 6               1                0              0             0             0
##   skin_colorunknown skin_colorwhite skin_coloryellow eye_colorblack
## 1                 0               0                0              0
## 2                 0               1                0              0
## 3                 0               0                0              0
## 4                 0               0                0              0
## 5                 0               0                0              0
## 6                 0               0                0              0
##   eye_colorblue eye_colorblue-gray eye_colorbrown eye_colorhazel
## 1             1                  0              0              0
## 2             0                  0              0              0
## 3             0                  0              1              0
## 4             1                  0              0              0
## 5             1                  0              0              0
## 6             0                  0              1              0
##   eye_colororange eye_colorred eye_coloryellow homeworldAlderaan
## 1               0            0               0                 0
## 2               0            0               1                 0
## 3               0            0               0                 1
## 4               0            0               0                 0
## 5               0            0               0                 0
## 6               0            0               0                 0
##   homeworldBespin homeworldCerea homeworldConcord Dawn homeworldCorellia
## 1               0              0                    0                 0
## 2               0              0                    0                 0
## 3               0              0                    0                 0
## 4               0              0                    0                 0
## 5               0              0                    0                 0
## 6               0              0                    0                 0
##   homeworldDathomir homeworldDorin homeworldEndor homeworldHaruun Kal
## 1                 0              0              0                   0
## 2                 0              0              0                   0
## 3                 0              0              0                   0
## 4                 0              0              0                   0
```

```
## 5                 0              0              0                    0
## 6                 0              0              0                    0
##   homeworldKamino homeworldKashyyyk homeworldMirial homeworldMon Cala
## 1               0                 0               0                 0
## 2               0                 0               0                 0
## 3               0                 0               0                 0
## 4               0                 0               0                 0
## 5               0                 0               0                 0
## 6               0                 0               0                 0
##   homeworldNaboo homeworldRyloth homeworldSerenno homeworldSocorro
## 1              0               0                0                0
## 2              0               0                0                0
## 3              0               0                0                0
## 4              0               0                0                0
## 5              0               0                0                0
## 6              0               0                0                0
##   homeworldStewjon homeworldTatooine homeworldTrandosha speciesCerean
## 1                0                 1                  0             0
## 2                0                 1                  0             0
## 3                0                 0                  0             0
## 4                0                 1                  0             0
## 5                0                 1                  0             0
## 6                0                 1                  0             0
##   speciesEwok speciesGungan speciesHuman speciesKel Dor speciesMirialan
## 1           0             0            1               0               0
## 2           0             0            1               0               0
## 3           0             0            1               0               0
## 4           0             0            1               0               0
## 5           0             0            1               0               0
## 6           0             0            1               0               0
##   speciesMon Calamari speciesTrandoshan speciesTwi'lek speciesWookiee
## 1                   0                 0              0              0
## 2                   0                 0              0              0
## 3                   0                 0              0              0
## 4                   0                 0              0              0
## 5                   0                 0              0              0
## 6                   0                 0              0              0
##   speciesZabrak sexfemale sexmale
## 1             0         0       1
## 2             0         0       1
## 3             0         1       0
## 4             0         0       1
## 5             0         1       0
## 6             0         0       1
```

b. Use SVM to predict gender and report the accuracy.

```
train_control_loocv = trainControl(method = "LOOCV", number = 10)
svm_starwars <- train(gender ~., data = starwars_dummies, method = "svmLinear",
              trControl = train_control_loocv)
```

```
## Warning in .local(x, ...): Variable(s) '' constant. Cannot scale data.
```

```
## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.
```

```
svm_starwars
```

```
## Support Vector Machines with Linear Kernel
##
## 29 samples
## 66 predictors
##  2 classes: 'feminine', 'masculine'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 28, 28, 28, 28, 28, 28, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9655172  0.9010239
##
```

```
## Tuning parameter 'C' was held constant at a value of 1
```

c. Given that we have so many variables, it makes sense to consider using PCA. Run PCA on the data and determine an appropriate number of components to use. Document how you made the decision, including any graphs you used. Create a reduced version of the data with that number of principle components. Note: make sure to remove gender from the data before running PCA because it would be cheating if PCA had access to the label you will use. Add it back in after reducing the data and show the result.

PCA only works with numerical values To get a list of potential predictors as a set of column indices we can use nearZeroVar function.

```
nzv <- nearZeroVar(starwars_dummies)
length(nzv)
```

```
## [1] 39
```

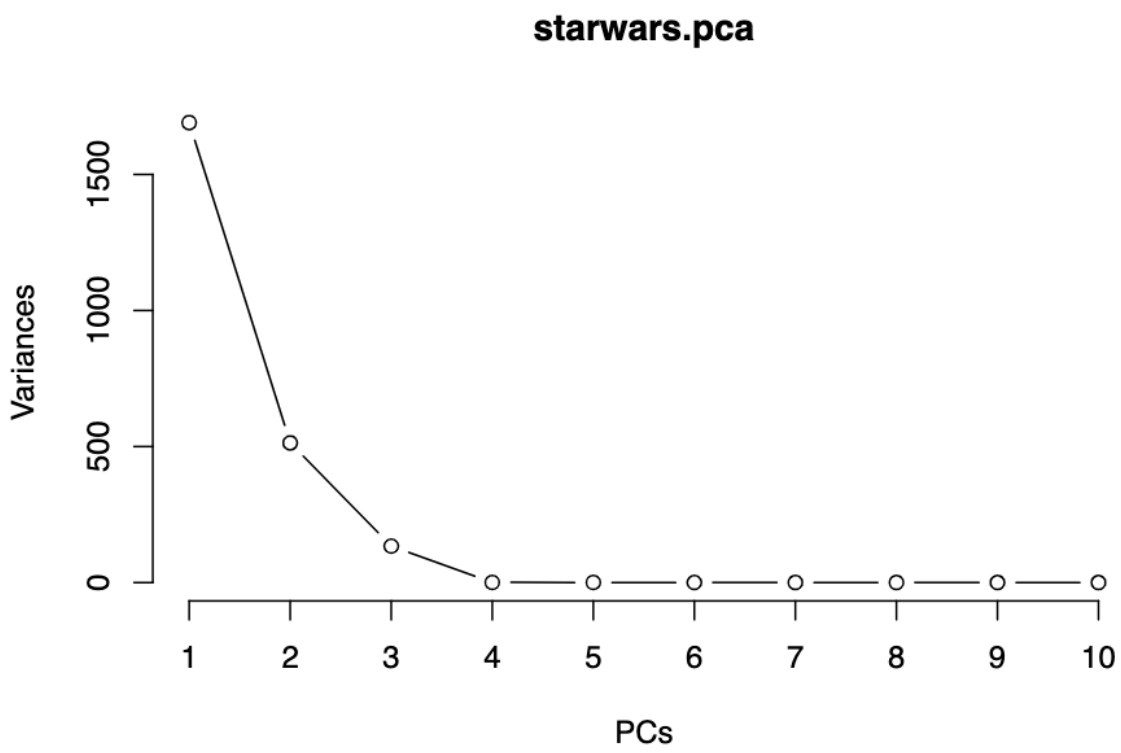
```
#Remove gender column
starwars_dummies_no_gender <- starwars_dummies %>% select(-c("gender"))
```

The Caret library version of the PCA requires an exact number components as a parameter. To determine this number we can use summary statistics and various visualizations available for the base R principal component function prcomp. Here the summary will give you the cumulative proportions of the PCs. Scree plot will give you similar information visually.

```
starwars.pca <- prcomp(starwars_dummies_no_gender)
summary(starwars.pca)
```

```
## Importance of components:
##                             PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     41.1152 22.6543 11.5834 0.78553 0.73967 0.59403 0.57852
## Proportion of Variance  0.7219  0.2192  0.0573 0.00026 0.00023 0.00015 0.00014
## Cumulative Proportion   0.7219  0.9411  0.9984 0.99863 0.99886 0.99901 0.99916
##                             PC8    PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.51620 0.4766 0.42229 0.38431 0.37487 0.36960 0.35334
## Proportion of Variance 0.00011 0.0001 0.00008 0.00006 0.00006 0.00006 0.00005
## Cumulative Proportion  0.99927 0.9994 0.99944 0.99951 0.99957 0.99963 0.99968
##                            PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.34330 0.32559 0.31189 0.29516 0.27246 0.25433 0.24201
## Proportion of Variance 0.00005 0.00005 0.00004 0.00004 0.00003 0.00003 0.00003
## Cumulative Proportion  0.99973 0.99977 0.99982 0.99985 0.99988 0.99991 0.99994
##                            PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     0.20171 0.19847 0.16261 0.15702 0.11978 0.03711 0.01792
## Proportion of Variance 0.00002 0.00002 0.00001 0.00001 0.00001 0.00000 0.00000
## Cumulative Proportion  0.99995 0.99997 0.99998 0.99999 1.00000 1.00000 1.00000
##                             PC29
## Standard deviation     3.836e-15
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```r
# Visualize the scree plot
screeplot(starwars.pca, type = "l") + title(xlab = "PCs")
```




```
## integer(0)
```

Looking at these results we can say that we capture most of the variance by using 3 principal components at +95% variance. Now, we can use 3 PCs to model our data. Caret package preProcess function here can give you a new dataset with the PCs instead of the predictors.

```r
target <- starwars %>% dplyr::select(gender)
preProc <- preProcess(starwars_dummies, method="pca", pcaComp=3)
starwars.pc <- predict(preProc, starwars_dummies)
starwars.pc$gender <- starwars$gender
head(starwars.pc)
```

```
##      gender        PC1       PC2        PC3
## 1 masculine -0.2119596 1.7798166 -0.1805155
## 2 masculine  2.6062413 0.7266943  0.0536266
## 3  feminine -3.5052503 0.3029799 -0.5375610
## 4 masculine  0.3975989 1.7973980  0.7166245
## 5  feminine -2.1157926 0.6934936  1.1160075
## 6 masculine -0.7175118 1.4161888 -0.5011268
```

d. Use SVM to predict gender again, but this time use the data resulting from PCA. Evaluate the results with a confusion matrix and at least two partitioning methods, using grid search on the C parameter each time.

```r
library(caret)
library(e1071)

set.seed(123)
index = createDataPartition(y=starwars.pc$gender, p=0.7, list=FALSE)
train_set = starwars.pc[index,]
test_set = starwars.pc[-index,]
svm_split <- train(gender ~., data = train_set, method = "svmLinear")
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```r
pred_split <- predict(svm_split, test_set)
sum(pred_split == test_set$gender) / nrow(test_set)
```

```
## [1] 0.8571429
```

```r
#Confusion matrix:
test_set$Sgender <- pred_split

# Convert the Sgender column to a factor with levels from pred_split
test_set$Sgender <- factor(test_set$Sgender, levels = levels(pred_split))

confusionMatrix(test_set$Sgender, pred_split)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  feminine masculine
##   feminine         2         0
##   masculine        0         5
##
##                Accuracy : 1
##                  95% CI : (0.5904, 1)
##     No Information Rate : 0.7143
##     P-Value [Acc > NIR] : 0.09486
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.2857
##          Detection Rate : 0.2857
##    Detection Prevalence : 0.2857
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : feminine
##
```

grid search:

```
train_control = trainControl(method = "cv", number = 10)
grid <- expand.grid(C = 10^seq(-5,2,0.5))

# Fit the model
svm_grid <- train(gender ~., data = starwars.pc, method = "svmLinear",
              trControl = train_control, tuneGrid = grid)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
# View grid search result
svm_grid
```

```
## Support Vector Machines with Linear Kernel
##
## 29 samples
##  3 predictor
##  2 classes: 'feminine', 'masculine'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 26, 27, 27, 26, 26, 25, ...
## Resampling results across tuning parameters:
##
##   C             Accuracy   Kappa
##   1.000000e-05  0.8166667  0.0000000
##   3.162278e-05  0.8166667  0.0000000
##   1.000000e-04  0.8166667  0.0000000
##   3.162278e-04  0.8166667  0.0000000
##   1.000000e-03  0.8166667  0.0000000
##   3.162278e-03  0.8166667  0.0000000
##   1.000000e-02  0.8166667  0.0000000
##   3.162278e-02  0.8166667  0.0000000
##   1.000000e-01  0.9000000  0.5000000
##   3.162278e-01  0.9000000  0.5000000
##   1.000000e+00  0.9166667  0.7714286
##   3.162278e+00  0.9500000  0.8571429
##   1.000000e+01  0.9500000  0.8571429
##   3.162278e+01  0.9500000  0.8571429
##   1.000000e+02  0.9500000  0.8571429
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 3.162278.
```

e. Whether or not it has improved the accuracy, what has PCA done for the complexity of the model?

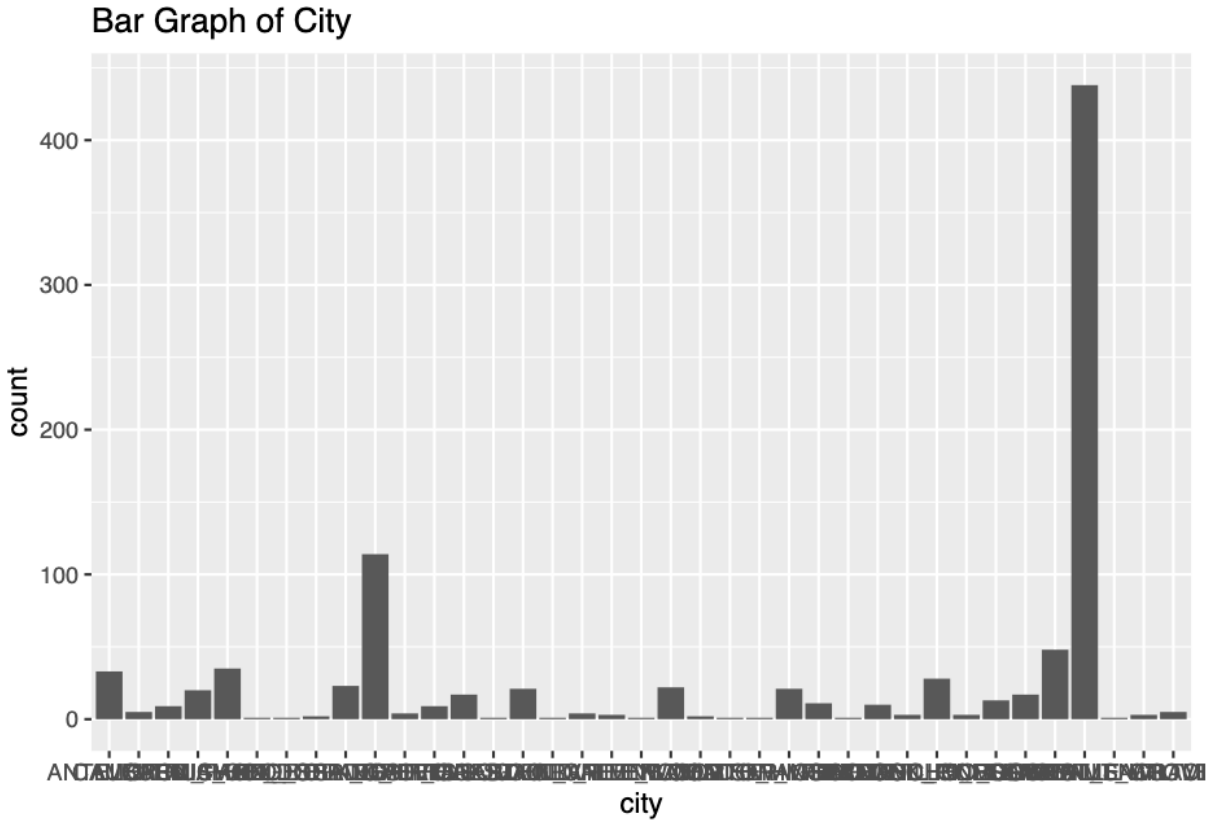
PCA has reduced the complexity of the model by reducing the number of predictors from 66 to 3 while maintaining a high level of accuracy. The original dataset had 66 predictors, which could lead to overfitting and increased computational complexity. By using PCA to reduce the dimensionality to 3 principal components, we have a simpler model that still explains over 95% of the variance in the data. This reduction in complexity can lead to a more interpretable model and potentially better generalization to new data.

Problem 4

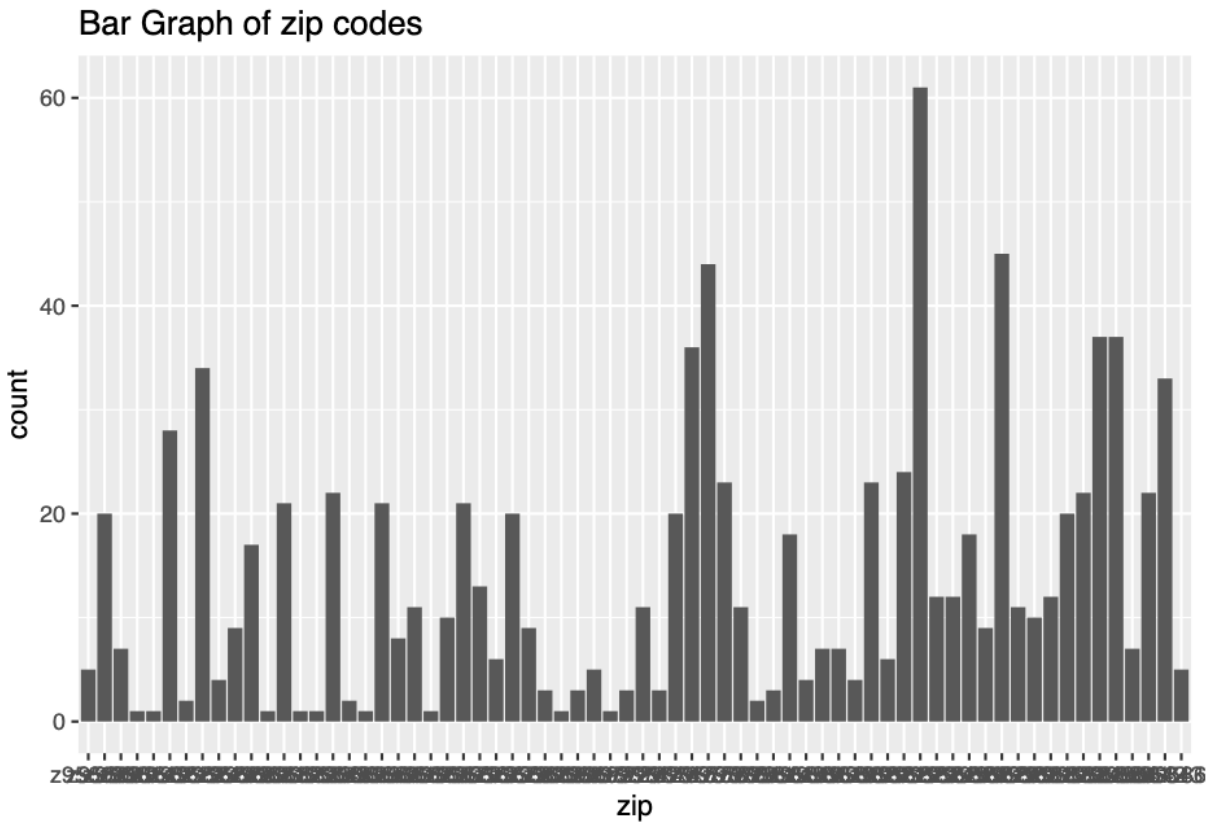
Use the Sacremento data from the caret library by running data(Sacremento) after loading caret. This data is about housing prices in Sacramento, California. Remove the zip and city variables. a. Explore the variables to see if they have reasonable distributions and show your work. We will be predicting the type variable – does that mean we have a class imbalance?

Exploring variables

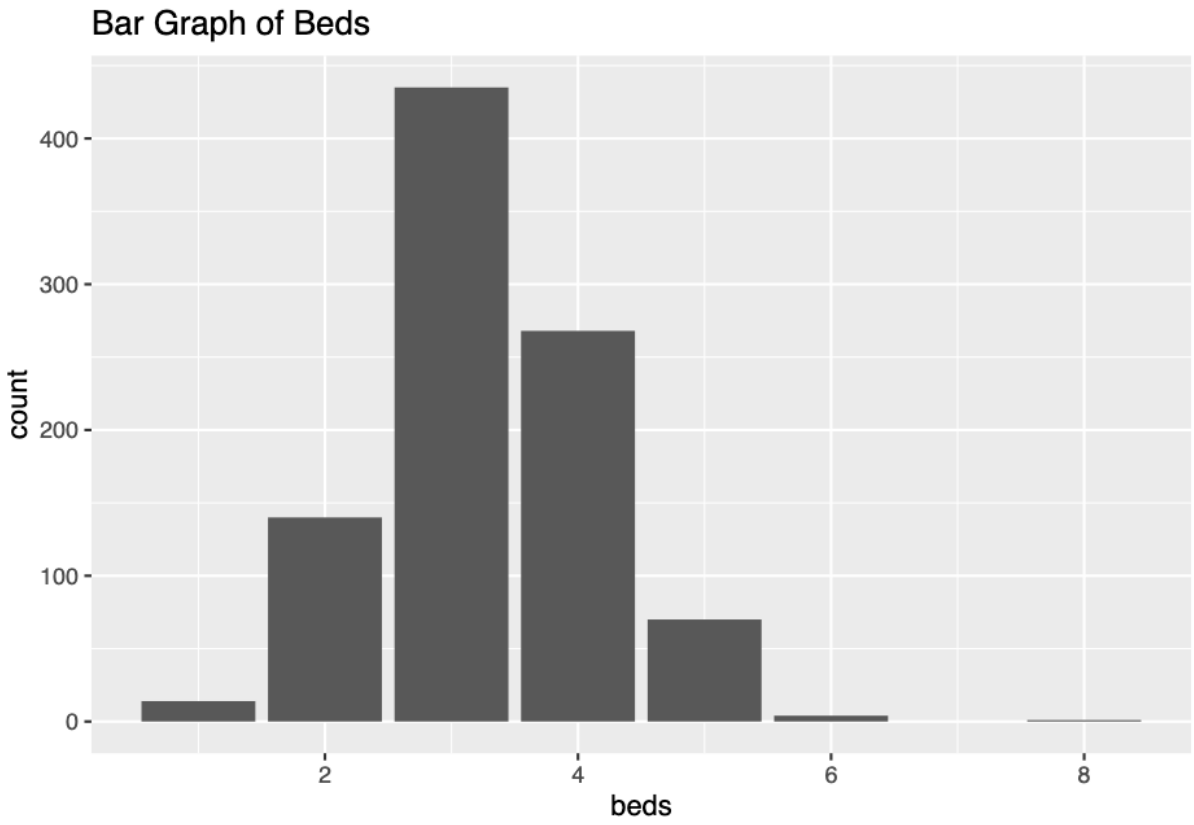
```
data("Sacramento")
library(ggplot2)
ggplot(Sacramento, aes(x=city)) + geom_bar()+ggtitle("Bar Graph of City")
```



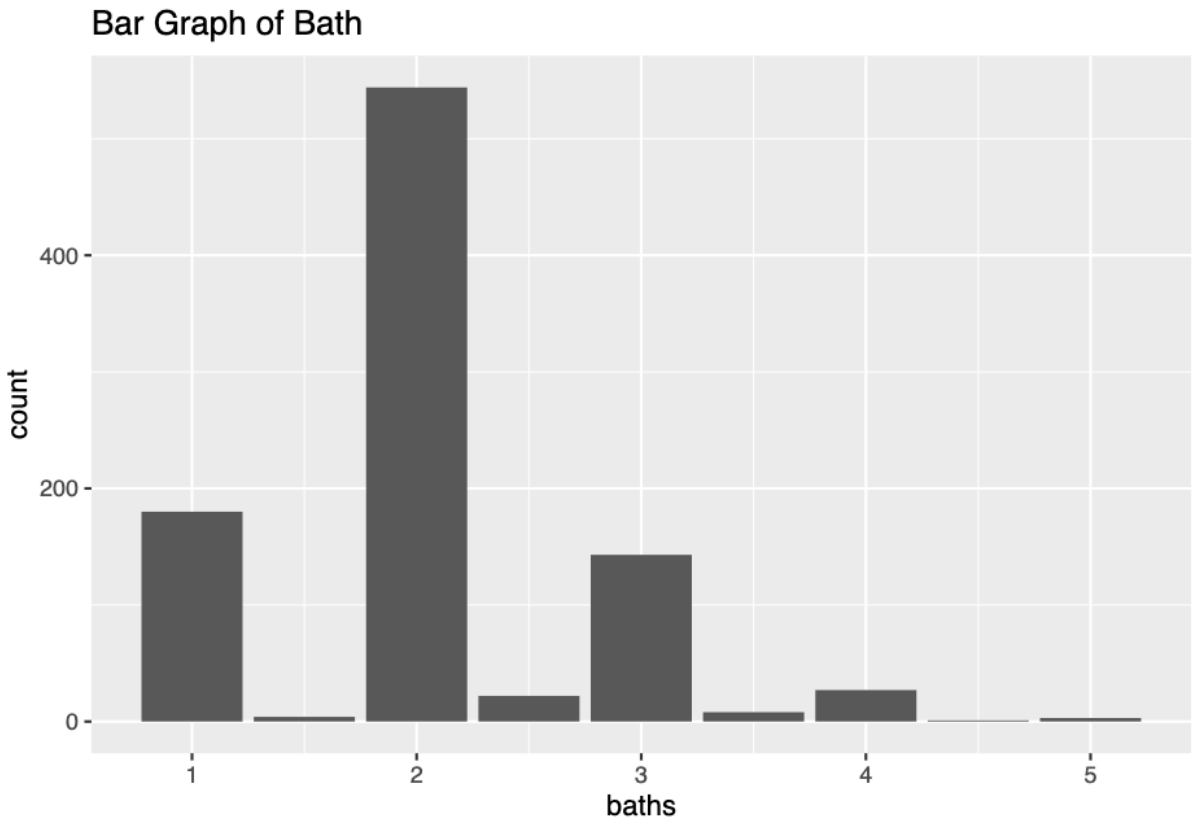

```
ggplot(Sacramento, aes(x=zip)) + geom_bar()+ggtitle("Bar Graph of zip codes")
```



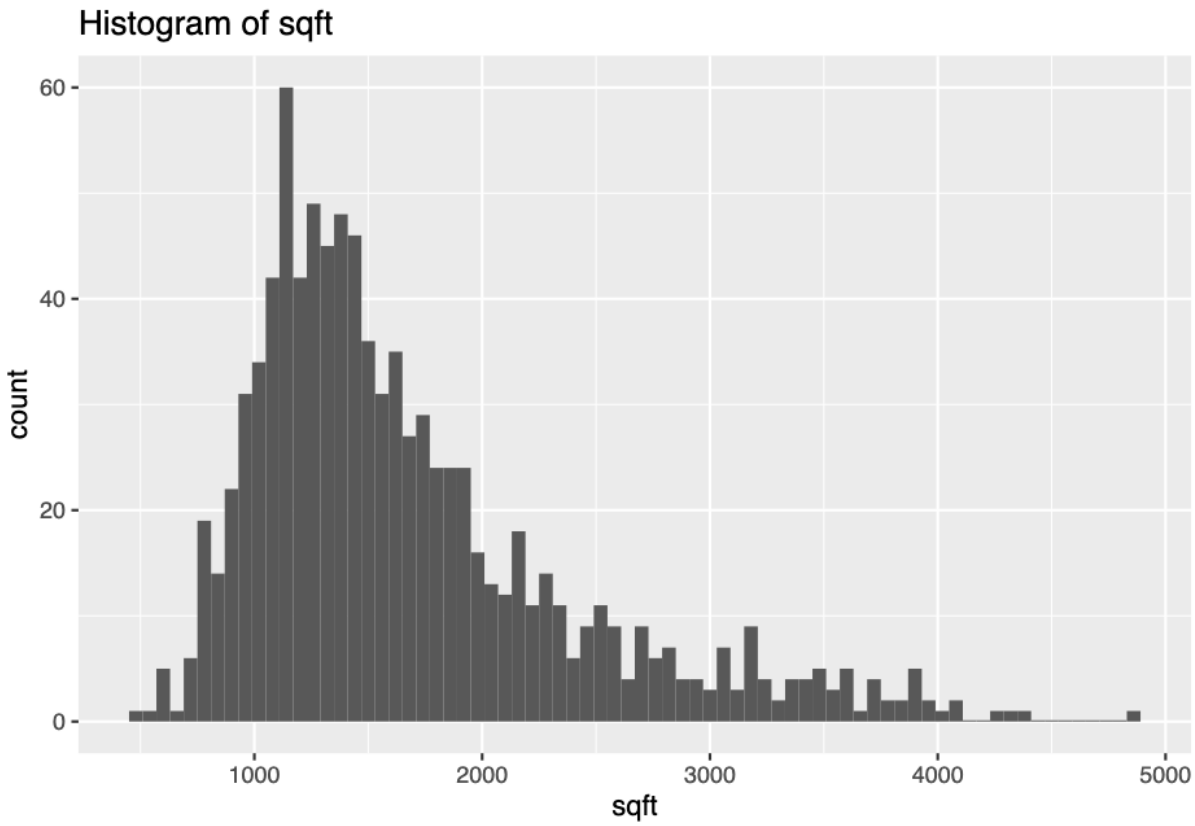

```
ggplot(Sacramento, aes(x=beds)) + geom_bar()+ggtitle("Bar Graph of Beds")
```



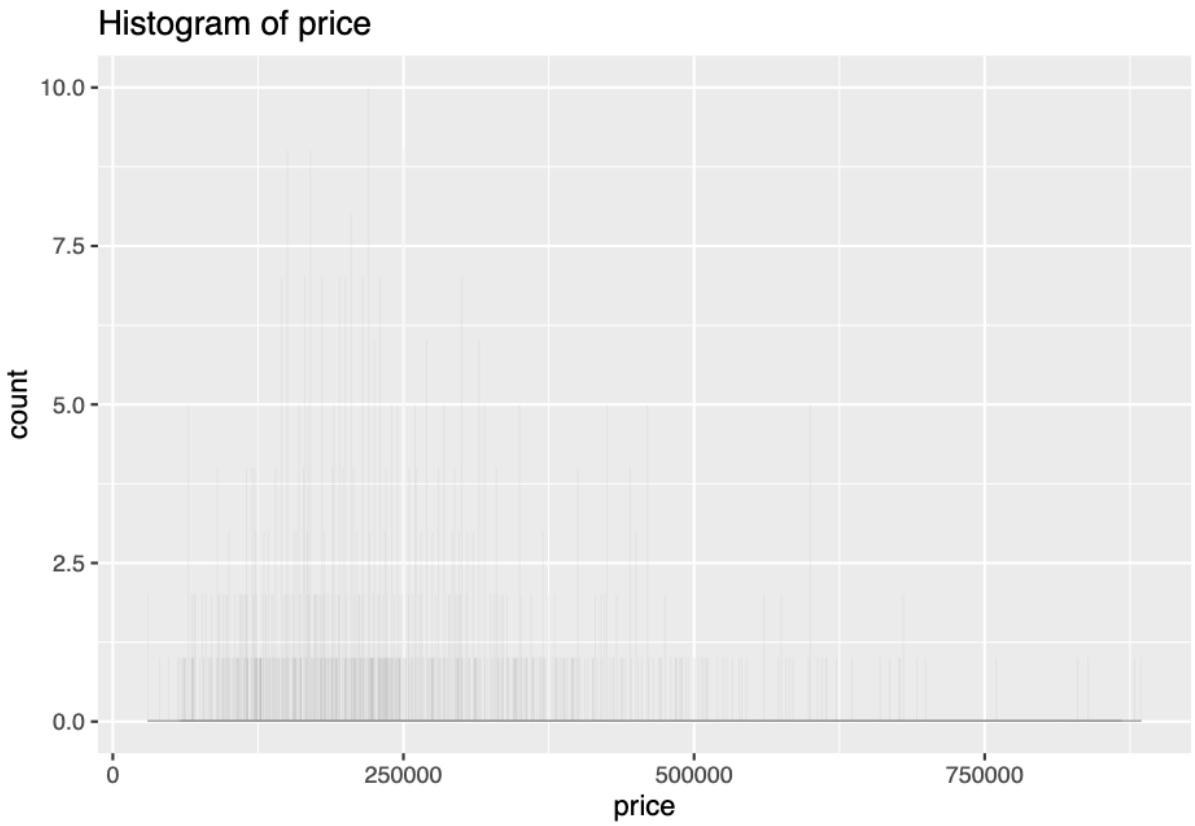

```
ggplot(Sacramento, aes(x=baths)) + geom_bar()+ggtitle("Bar Graph of Bath")
```



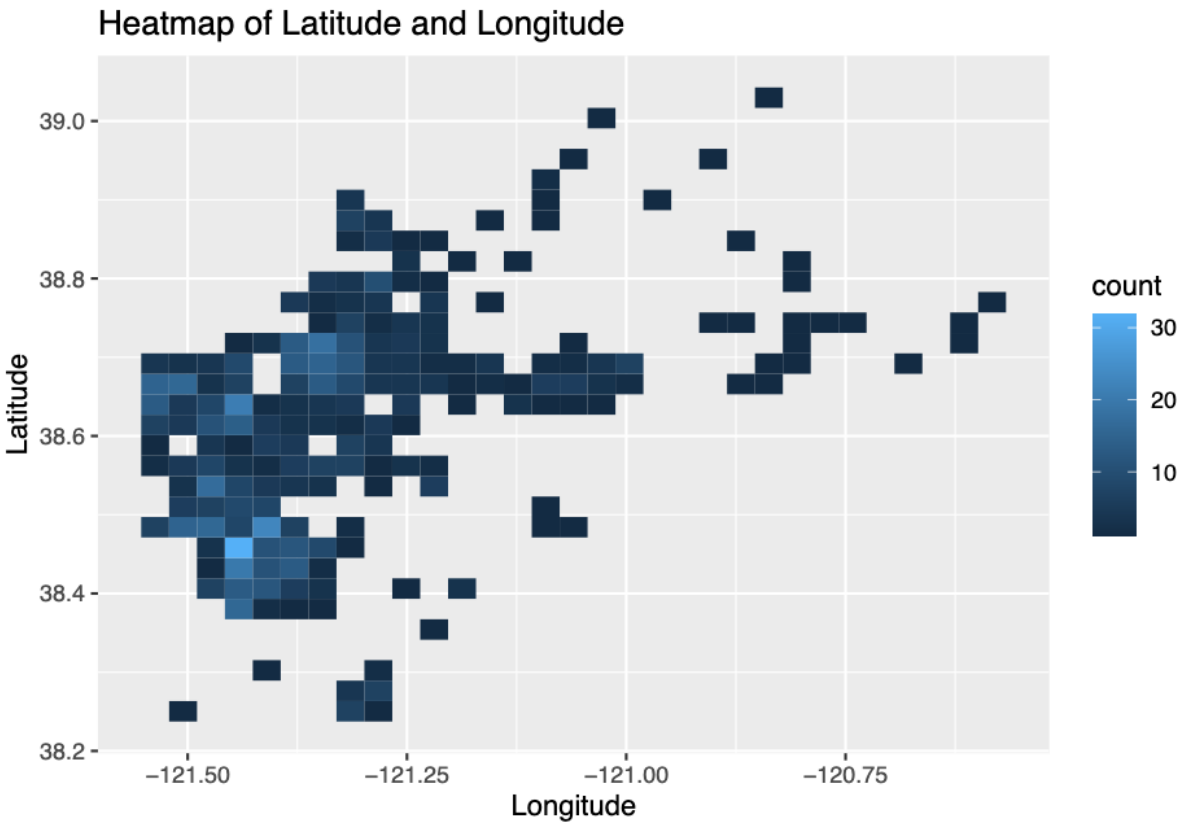

```
ggplot(Sacramento, aes(sqft)) + geom_histogram(binwidth = 60)+ggtitle("Histogram of sqft")
```




```
ggplot(Sacramento, aes(price)) + geom_histogram(binwidth = 60)+ggtitle("Histogram of price")
```

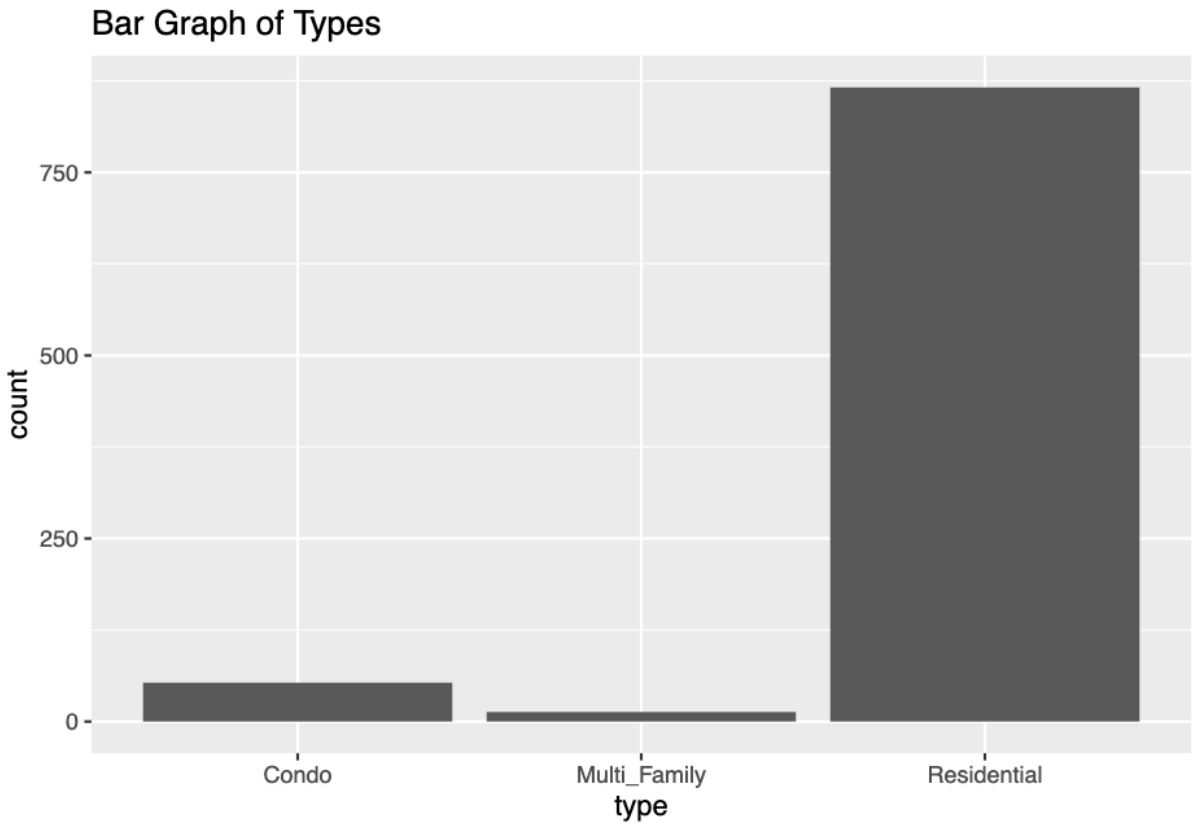



```
ggplot(Sacramento, aes(x = longitude, y = latitude)) + geom_bin2d() + labs(x = "Longitude", y = "Latitu
```



Class imbalance: In the type variable-Residential has more values than condo and multi-family

```
ggplot(Sacramento, aes(x=type)) + geom_bar()+ggtitle("Bar Graph of Types")
```



b. There are lots of options for working on the data to try to improve the performance of SVM, including (1) removing other variables that you know should not be part of the prediction, (2) dealing with extreme variations in some variables with smoothing, normalization or a log transform, (3) applying PCA, and (4) to removing outliers. Pick one now and continue.

(1) removing other variables that you know should not be part of the prediction From the Sacramento dataset, it seems like the variables "city", "zip", "latitude", and "longitude" are not directly related to the prediction of the property type ("type"). These variables are more about the location and identification of the property, which might not be relevant for predicting its type. Therefore, I would consider removing these variables if they are not required for my prediction task.

```
Sacramento<- na.omit(Sacramento)
Sacramento_cleaned <- Sacramento %>% select(-c("city", "zip", "latitude","longitude"))
head(Sacramento_cleaned)
```

```
##   beds baths sqft        type price
## 1    2     1  836 Residential 59222
## 2    3     1 1167 Residential 68212
## 3    2     1  796 Residential 68880
## 4    2     1  852 Residential 69307
## 5    2     1  797 Residential 81900
## 6    3     1 1122       Condo 89921
```

c. Use SVM to predict type and use grid search to get the best accuracy you can. The accuracy may be good, but look at the confusion matrix as well. Report what you find. Note that the kappa value

provided with your SVM results can also help you see this. It is a measure of how well the classifier performed that takes into account the frequency of the classes.

```
# Assuming 'Sacramento_cleaned' is your dataset

set.seed(123)
index = createDataPartition(y = Sacramento_cleaned$type, p = 0.7, list = FALSE)
train_set = Sacramento_cleaned[index, ]
test_set = Sacramento_cleaned[-index, ]

# Fit the SVM model using the training set
svm_model <- train(type ~., data = train_set, method = "svmLinear")

# Make predictions on the test set
pred <- predict(svm_model, test_set)

# Calculate accuracy
accuracy <- sum(pred == test_set$type) / nrow(test_set)
accuracy
```

```
## [1] 0.9350181
```

```
# Generate confusion matrix
confusionMatrix(test_set$type, pred)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Condo Multi_Family Residential
##   Condo             0            0          15
##   Multi_Family      0            0           3
##   Residential       0            0         259
##
## Overall Statistics
##
##                Accuracy : 0.935
##                  95% CI : (0.8992, 0.961)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Condo Class: Multi_Family Class: Residential
## Sensitivity                    NA                  NA              0.935
## Specificity               0.94585             0.98917                 NA
## Pos Pred Value                 NA                  NA                 NA
## Neg Pred Value                 NA                  NA                 NA
## Prevalence                0.00000             0.00000              1.000
## Detection Rate            0.00000             0.00000              0.935
```

```
## Detection Prevalence      0.05415              0.01083              0.935
## Balanced Accuracy             NA                   NA                 NA
```

The overall accuracy of the model is 0.935, indicating that it correctly predicts the property type 93.5% of the time. The model shows a high specificity for predicting Condo (0.94585) and Multi_Family (0.98917) properties, meaning it is good at correctly identifying these types when they are the actual class. However, there is an issue with class imbalance, particularly with the Condo and Multi_Family classes, where the model fails to predict any instances correctly. This imbalance can skew the accuracy metrics and make it seem like the model performs better than it actually does. For the Residential class, the sensitivity (true positive rate) is 0.935, indicating that the model is fairly good at identifying Residential properties when they are present. It's important to note that the Precision, Recall, and F1-score metrics are not calculated due to the class imbalance issue. Overall, while the model shows good accuracy, it may not be reliable for predicting Condo and Multi_Family properties due to the class imbalance. Further adjustments, such as resampling techniques or using different algorithms, may be necessary to improve the model's performance for these classes.

Grid search:

```
train_control = trainControl(method = "cv", number = 10)
grid <- expand.grid(C = 10^seq(-5,2,0.5))

# Fit the model
svm_grid_sc <- train(type ~., data = Sacramento_cleaned, method = "svmLinear",
                     trControl = train_control, tuneGrid = grid)
 # View grid search result
 svm_grid_sc
```

```
## Support Vector Machines with Linear Kernel
##
## 932 samples
##   4 predictor
##   3 classes: 'Condo', 'Multi_Family', 'Residential'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 839, 838, 839, 839, 839, 840, ...
## Resampling results across tuning parameters:
##
##   C             Accuracy   Kappa
##   1.000000e-05  0.9292352   0.000000000
##   3.162278e-05  0.9292352   0.000000000
##   1.000000e-04  0.9292352   0.000000000
##   3.162278e-04  0.9292352   0.000000000
##   1.000000e-03  0.9292352   0.000000000
##   3.162278e-03  0.9292352   0.000000000
##   1.000000e-02  0.9292352   0.000000000
##   3.162278e-02  0.9292352   0.000000000
##   1.000000e-01  0.9292352   0.000000000
##   3.162278e-01  0.9270846  -0.003047091
##   1.000000e+00  0.9302876   0.068890770
##   3.162278e+00  0.9302876   0.068890770
##   1.000000e+01  0.9302876   0.068890770
##   3.162278e+01  0.9302876   0.068890770
##   1.000000e+02  0.9302876   0.068890770
```

```
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 1.
```

d. Return to (b) and try at least one other way to try to improve the data before running SVM again, as in (c).

Apply PCA:

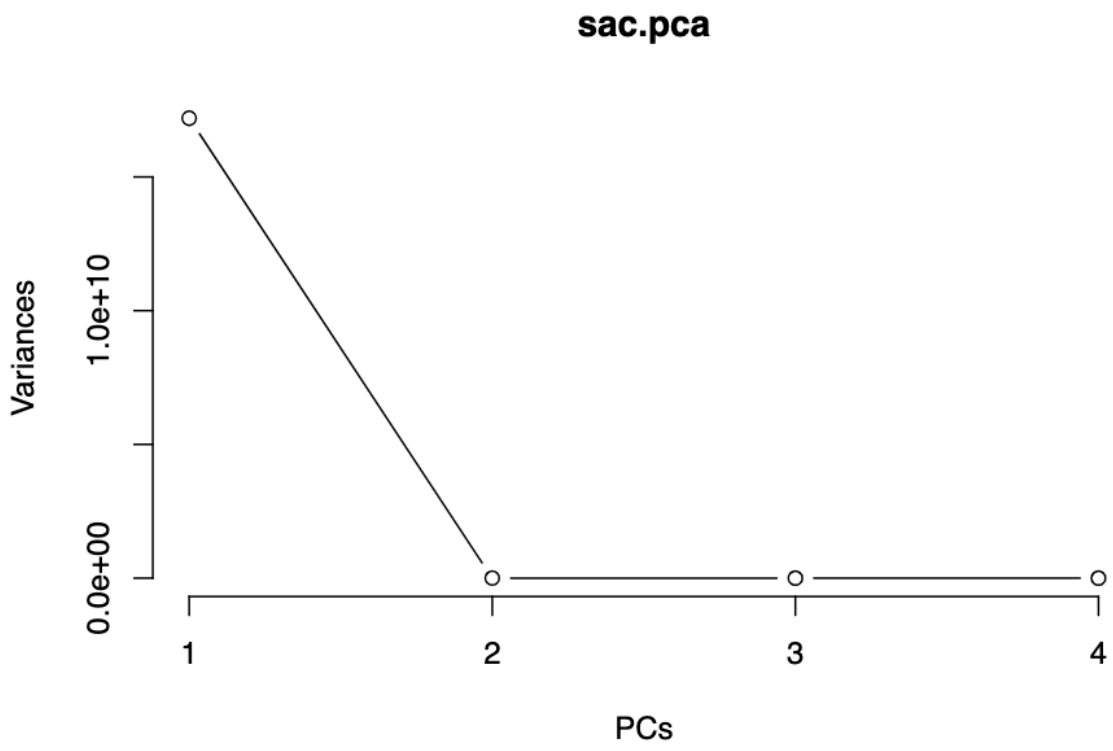
```
nzv <- nearZeroVar(Sacramento)
length(nzv)
```

```
## [1] 0
```

```
#remove type from Sacramento
sac_no_type <- Sacramento_cleaned %>% select(-c("type"))
#apply pca
sac.pca <- prcomp(sac_no_type)
summary(sac.pca)
```

```
## Importance of components:
##                           PC1       PC2    PC3    PC4
## Standard deviation     131128 465.68940 0.6312 0.4379
## Proportion of Variance      1   0.00001 0.0000 0.0000
## Cumulative Proportion       1   1.00000 1.0000 1.0000
```

```
#scree plot
screeplot(sac.pca, type = "l") + title(xlab = "PCs")
```




```
## integer(0)
```

Looking at these results we can say that we capture most of the variance by using 1 principal component at +95% variance. Now, we can use 1 PC to model our data. Caret package preProcess function here can give you a new dataset with the PCs instead of the predictors.

```
target <- Sacramento %>% dplyr::select(type)
preProc <- preProcess(Sacramento_cleaned, method="pca", pcaComp=1)
sac.pc <- predict(preProc, Sacramento_cleaned)
# Put back target column
sac.pc$type <- Sacramento$type
head(sac.pc)
```

```
##          type       PC1
## 1 Residential -2.724898
## 2 Residential -1.908414
## 3 Residential -2.720446
## 4 Residential -2.676714
## 5 Residential -2.673052
## 6       Condo -1.864562
```

SVM:

```r
set.seed(123)
index = createDataPartition(y=sac.pc$type, p=0.7, list=FALSE)
train_set = sac.pc[index,]
test_set = sac.pc[-index,]
svm_split <- train(type ~., data = train_set, method = "svmLinear")
pred_split <- predict(svm_split, test_set)
sum(pred_split == test_set$type) / nrow(test_set)
```

```
## [1] 0.9350181
```

```r
#Confusion matrix:
test_set$Stype <- pred_split

# Convert the Stype column to a factor with levels from pred_split
test_set$Stype <- factor(test_set$Stype, levels = levels(pred_split))

confusionMatrix(test_set$Stype, pred_split)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Condo Multi_Family Residential
##   Condo             0            0           0
##   Multi_Family      0            0           0
##   Residential       0            0         277
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9868, 1)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : NaN
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Condo Class: Multi_Family Class: Residential
## Sensitivity                    NA                  NA                  1
## Specificity                     1                   1                 NA
## Pos Pred Value                 NA                  NA                 NA
## Neg Pred Value                 NA                  NA                 NA
## Prevalence                      0                   0                  1
## Detection Rate                  0                   0                  1
## Detection Prevalence            0                   0                  1
## Balanced Accuracy              NA                  NA                 NA
```

In the confusion matrix generated for the prediction of property types in the Sacramento dataset, there is a significant class imbalance. The dataset consists of 277 instances of residential properties and no instances of condos or multi-family properties in the test set. This class imbalance can lead to challenges in model

evaluation, as the model may perform well in predicting the majority class (residential) but may struggle with the minority classes (condo and multi-family).

e. In the end, some data are just so imbalanced that a classifier is never going to predict the minority class. Dealing with this is a huge topic. One simple possibility is to conclude that we do not have enough data to support predicting the very infrequent class(es) and remove them. If they are not actually important to the reason we are making the prediction, that could be fine. Another approach is to force the data to be more even by sampling.

Create a copy of the data that includes all the data from the two smaller classes, plus a small random sample of the large class (you can do this by separating those data with a filter, sampling, then attaching them back on). Check the distributions of the variables in this new data sample to make sure they are reasonably close to the originals using visualization and/or summary statistics. We want to make sure we did not get a strange sample where everything was cheap or there were only studio apartments, for example. You can rerun the sampling a few times if you are getting strange results. If it keeps happening, check your process. Use SVM to predict type one this new, more balanced dataset and report its performance with a confusion matrix and with grid search to get the best accuracy.

```
# Filter the data
condo_data <- Sacramento_cleaned %>% filter(type == "Condo")
multi_family_data <- Sacramento_cleaned %>% filter(type == "Multi_Family")
residential_data <- Sacramento_cleaned %>% filter(type == "Residential")

# Sample the residential_data
set.seed(123) # for reproducibility
sample_residential_data <- residential_data %>% sample_n(size = nrow(condo_data) + nrow(multi_family_da

# Combine the sampled residential_data with condo_data and multi_family_data
balanced_data <- rbind(condo_data, multi_family_data, sample_residential_data)

# Check the distributions
summary(balanced_data)
```

```
##       beds           baths            sqft                  type
##  Min.   :1.000   Min.   :1.000   Min.   : 484.0   Condo       :53
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.: 991.5   Multi_Family:13
##  Median :3.000   Median :2.000   Median :1258.5   Residential :66
##  Mean   :2.856   Mean   :1.992   Mean   :1438.3
##  3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:1692.0
##  Max.   :8.000   Max.   :5.000   Max.   :4090.0
##      price
##  Min.   : 40000
##  1st Qu.:118312
##  Median :183916
##  Mean   :207362
##  3rd Qu.:251614
##  Max.   :680000
```

SVM to predict the class

```
# Split the data into training and testing sets
index <- createDataPartition(y = balanced_data$type, p = 0.7, list = FALSE)
train_set <- balanced_data[index, ]
```

```r
test_set <- balanced_data[-index, ]

# Train the SVM model
svm_model <- train(type ~ ., data = train_set, method = "svmLinear")

# Predict on the test set
pred <- predict(svm_model, test_set)

# Confusion matrix
confusionMatrix(test_set$type, pred)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Condo Multi_Family Residential
##   Condo            14            0           1
##   Multi_Family      1            1           1
##   Residential       2            0          17
##
## Overall Statistics
##
##                Accuracy : 0.8649
##                  95% CI : (0.7123, 0.9546)
##     No Information Rate : 0.5135
##     P-Value [Acc > NIR] : 7.672e-06
##
##                   Kappa : 0.7533
##
##  Mcnemar's Test P-Value : 0.5062
##
## Statistics by Class:
##
##                      Class: Condo Class: Multi_Family Class: Residential
## Sensitivity                0.8235             1.00000              0.8947
## Specificity                0.9500             0.94444              0.8889
## Pos Pred Value             0.9333             0.33333              0.8947
## Neg Pred Value             0.8636             1.00000              0.8889
## Prevalence                 0.4595             0.02703              0.5135
## Detection Rate             0.3784             0.02703              0.4595
## Detection Prevalence       0.4054             0.08108              0.5135
## Balanced Accuracy          0.8868             0.97222              0.8918
```

```r
# Grid search for best accuracy
grid <- expand.grid(C = 10^seq(-5, 2, 0.5))
svm_grid <- train(type ~ ., data = train_set, method = "svmLinear",
                  trControl = trainControl(method = "cv", number = 5),
                  tuneGrid = grid)

# Best model
best_model <- svm_grid$bestTune
best_model
```

```
##           C
```

```
## 10 0.3162278
```

Problem 5

To understand just how much different subsets can differ, create a 5 fold partitioning of the cars data included in R (mtcars) and visualize the distribution of the gears variable across the folds. Rather than use the fancy trainControl methods for making the folds, create them directly so you actually can keep track of which data points are in which fold. This is not covered in the tutorial, but it is quick. Here is code to create 5 folds and a variable in the data frame that contains the fold index of each point. Use that resulting data frame to create your visualization. mycars <- mtcars # make a copy to modify mycars$folds = 0 # initialize new variable to hold fold indices

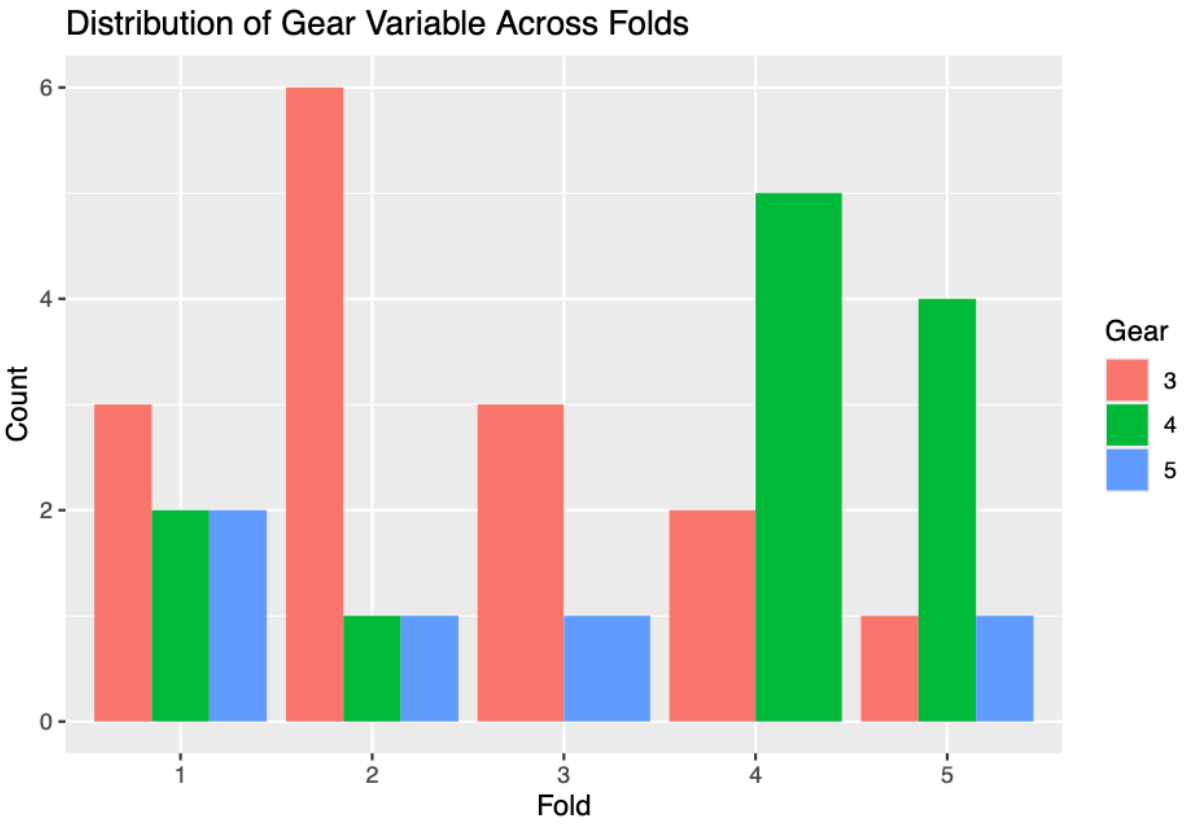
```r
# Load the mtcars dataset
data(mtcars)

# Make a copy of the dataset
mycars <- mtcars
mycars$folds <- 0

# Create 5 folds for cross-validation
flds <- createFolds(1:nrow(mycars), k = 5, list = TRUE)

# Assign fold indices to the folds variable
for (i in 1:5) {
  mycars$folds[flds[[i]]] <- i
}

# Visualize the distribution of the gear variable across folds
library(ggplot2)
ggplot(mycars, aes(x = factor(folds), fill = factor(gear))) +
  geom_bar(position = "dodge") +
  labs(x = "Fold", y = "Count", fill = "Gear") +
  ggtitle("Distribution of Gear Variable Across Folds")
```


Distribution of Gear Variable Across Folds
Count
6
4
2
0
1
2
3
4
5
Fold
Gear
3
4
5