# Missing Data_Breast Cancer

The breast cancer data set breast-cancer-wisconsin.data.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin has missing values. 1. Use the mean/mode imputation method to impute values for the missing data. 2. Use regression to impute values for the missing data 3. Use regression with perturbation to impute values for the missing data. 4. use classification models, SVM and KNN to impute missing value.

```
# Find the missing data
rm(list = ls())
set.seed(1)
data <- read.table("breast-cancer-wisconsin.txt", stringsAsFactors = FALSE, header = FALSE, sep = ",")
for (i in 2:11) {
  print(paste0("V",i))
  print(table(data[,i]))
}
```

```
## [1] "V2"
##
##   1   2   3   4   5   6   7   8   9  10
## 145  50 108  80 130  34  23  46  14  69
## [1] "V3"
##
##   1   2   3   4   5   6   7   8   9  10
## 384  45  52  40  30  27  19  29   6  67
## [1] "V4"
##
##   1   2   3   4   5   6   7   8   9  10
## 353  59  56  44  34  30  30  28   7  58
## [1] "V5"
##
##   1   2   3   4   5   6   7   8   9  10
## 407  58  58  33  23  22  13  25   5  55
## [1] "V6"
##
##   1   2   3   4   5   6   7   8   9  10
##  47 386  72  48  39  41  12  21   2  31
## [1] "V7"
##
##   ?   1  10   2   3   4   5   6   7   8   9
##  16 402 132  30  28  19  30   4   8  21   9
## [1] "V8"
##
##   1   2   3   4   5   6   7   8   9  10
## 152 166 165  40  34  10  73  28  11  20
## [1] "V9"
##
```

```
##   1   2   3   4   5   6   7   8   9  10
## 443  36  44  18  19  22  16  24  16  61
## [1] "V10"
##
##   1   2   3   4   5   6   7   8  10
## 579  35  33  12   6   3   9   8  14
## [1] "V11"
##
##   2   4
## 458 241
```

```
data[which(data$V7 == "?"),]
```

```
##          V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11
## 24  1057013  8  4  5  1  2  ?  7  3   1   4
## 41  1096800  6  6  6  9  6  ?  7  8   1   2
## 140 1183246  1  1  1  1  1  ?  2  1   1   2
## 146 1184840  1  1  3  1  2  ?  2  1   1   2
## 159 1193683  1  1  2  1  3  ?  1  1   1   2
## 165 1197510  5  1  1  1  2  ?  3  1   1   2
## 236 1241232  3  1  4  1  2  ?  3  1   1   2
## 250  169356  3  1  1  1  2  ?  3  1   1   2
## 276  432809  3  1  3  1  2  ?  2  1   1   2
## 293  563649  8  8  8  1  2  ?  6 10   1   4
## 295  606140  1  1  1  1  2  ?  2  1   1   2
## 298   61634  5  4  3  1  2  ?  2  3   1   2
## 316  704168  4  6  5  6  7  ?  4  9   1   2
## 322  733639  3  1  1  1  2  ?  3  1   1   2
## 412 1238464  1  1  1  1  1  ?  2  1   1   2
## 618 1057067  1  1  1  1  1  ?  1  1   1   2
```

```
nrow(data[which(data$V7 == "?"),])/nrow(data)
```

```
## [1] 0.02288984
```

```
missing <- which(data$V7 == "?", arr.ind = TRUE)
missing
```

```
##  [1]  24  41 140 146 159 165 236 250 276 293 295 298 316 322 412 618
```

## Mean/Mode Imputation

```
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

# Find the mode of V7.
mode_V7 <- as.numeric(getmode(data[-missing,"V7"]))
mode_V7
```

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

```
# Impute V7 for observations with missing data for V7 to mode_V7.
data_mode_imp <- data
data_mode_imp[missing,]$V7 <- mode_V7
data_mode_imp$V7 <- as.integer(data_mode_imp$V7)
data_mode_imp$V7
```

```
##   [1]  1 10  2  4  1 10 10  1  1  1  1  1  3  3  9  1  1  1 10  1 10  7  1  1  1
##  [26]  7  1  1  1  1  1  1  5  1  1  1  1  1 10  7  1  3 10  1  1  1  9  1  1  8
##  [51]  3  4  5  8  8  5  6  1 10  2  3  2  8  2  1  2  1 10  9  1  1  2  1 10  4
##  [76]  2  1  1  3  1  1  1  1  2  9  4  8 10  1  1  1  1  1  1  1  1  1  1  6 10
## [101]  5  5  1  3  1  3 10 10  1  9  2  9 10  8  3  5  2 10  3  2  1  2 10 10  7
## [126]  1 10  1 10  1  1  1 10  1  1  2  1  1  1  1  1  1  5  5  1  1  8  2  1 10
## [151]  1 10  5  3  1 10  1  1  1 10 10  1  1  3  1  2 10  1  1  1  1  1  1 10 10
## [176] 10  1  1  1 10  1  1  1 10 10  1  8 10  8  1  8 10  1  1  1  1  7  1  1  1
## [201] 10 10  1  1  1 10  5  1  1  1 10  8  1 10 10  5  1  1  4  1  1 10  5  8 10
## [226]  1 10  5  1 10  7  8  1 10  1  1 10  2  9 10  2  1  1  5  1  2 10  9  1  1
## [251]  1 10 10 10  8 10  1  1  1  8 10 10 10 10  3  1 10 10  4  1 10  1 10  4  1
## [276]  1  1  1  1  7  1  1 10 10 10 10 10  1  5 10  1  1  1 10  1 10  5  1  1 10
## [301]  4  1 10  1 10 10  1  1  3  5  1  1  1  1  1  1 10  8  1  5 10  1  1 10  1
## [326]  1 10  1  4 10  8  1  1 10 10  1 10  1  1 10 10  1  1  1 10  1  1  1  1  8
## [351]  1  1  3 10  1  1  3 10  4  7 10 10  3  3  1  1 10 10  1  1  1  1  1  1  1
## [376]  1  1  1  1  1  1 10  1  1  1  1 10  1  1  2  1 10  1  1  1  1  1  1  1  1
## [401]  9  1  1  4  1  1  1  1  2  1  1  1  4  1 10  3 10  1  2  1  3 10  1  1  1
## [426] 10  1  2  1  1  1  1  1  1  8 10  1  1  1  1 10  4  3  2  1  1  1  1  1 10
## [451]  1  1  1 10  1  6 10  3  1  1  1  5  1  1  1  4 10 10  1  1  1  1  1  1  1
## [476]  1  1  1  1 10  1  1  5 10  1  3  1 10  3  4  1 10  1 10  5  1  1  1  1  1
## [501]  1  1  1  1  1  1  5  4  1  1  1  1  1  1 10 10  1  1  1 10  1  1  5 10  1
## [526]  1  1  1  1  1 10  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1 10  1  1  5
## [551]  1  1  1  5  1  1  1  1  1  1  1  1  1  1  1 10  1  3 10  5 10 10  1  1  2
## [576]  1  1  1  1  1  1 10 10  1  1  1 10  1  3  1  1 10 10  1 10  1  1  1  1  1
## [601]  1  1  1  1 10  8  1  1 10  1 10  2 10  1  1  1  1  1  1  1  1  2  1  1  1
## [626]  4  6  5  1  1  1  1  1  3  1  1  1  2  1  1  1  1  1  1  1  1  1  1  2  1
## [651]  4  1  1  1  1  1  1  1 10  1  1  1  1  1  1  1  1  1  1  5  8  1  1  1  1
## [676]  1  1  1  1  1 10 10  1  1  1  1  1  1  1  1  1  5  1  1  2  1  3  4  5
```

## Regression Imputation

```
data_modified <- data[-missing,2:10]
data_modified$V7 <- as.integer(data_modified$V7)
# Generate linear model using all other factors as predictors
model <- lm(V7~V2+V3+V4+V5+V6+V8+V9+V10, data = data_modified)
summary(model)
```

```
##
## Call:
## lm(formula = V7 ~ V2 + V3 + V4 + V5 + V6 + V8 + V9 + V10, data = data_modified)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -9.7316 -0.9426 -0.3002  0.6725  8.6998
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.616652   0.194975  -3.163  0.00163 **
## V2           0.230156   0.041691   5.521 4.83e-08 ***
## V3          -0.067980   0.076170  -0.892  0.37246
## V4           0.340442   0.073420   4.637 4.25e-06 ***
## V5           0.339705   0.045919   7.398 4.13e-13 ***
## V6           0.090392   0.062541   1.445  0.14883
## V8           0.320577   0.059047   5.429 7.91e-08 ***
## V9           0.007293   0.044486   0.164  0.86983
## V10         -0.075230   0.059331  -1.268  0.20524
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.274 on 674 degrees of freedom
## Multiple R-squared:  0.615,  Adjusted R-squared:  0.6104
## F-statistic: 134.6 on 8 and 674 DF,  p-value: < 2.2e-16
```

```
#  use stepwise for variable selection.
step(model,trace=0)
```

```
##
## Call:
## lm(formula = V7 ~ V2 + V4 + V5 + V8, data = data_modified)
##
## Coefficients:
## (Intercept)           V2           V4           V5           V8
##     -0.5360       0.2262       0.3173       0.3323       0.3238
```

```
# Generate the reduce model
model2 <- lm(V7~V2+V4+V5+V8, data = data_modified)
summary(model2)
```

```
##
## Call:
## lm(formula = V7 ~ V2 + V4 + V5 + V8, data = data_modified)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8115 -0.9531 -0.3111  0.6678  8.6889
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.53601    0.17514  -3.060   0.0023 **
## V2           0.22617    0.04121   5.488 5.75e-08 ***
## V4           0.31729    0.05086   6.239 7.76e-10 ***
## V5           0.33227    0.04431   7.499 2.03e-13 ***
## V8           0.32378    0.05606   5.775 1.17e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.274 on 678 degrees of freedom
## Multiple R-squared:  0.6129, Adjusted R-squared:  0.6107
## F-statistic: 268.4 on 4 and 678 DF,  p-value: < 2.2e-16
```

```
# cross-validation to test model
library(DAAG)
```

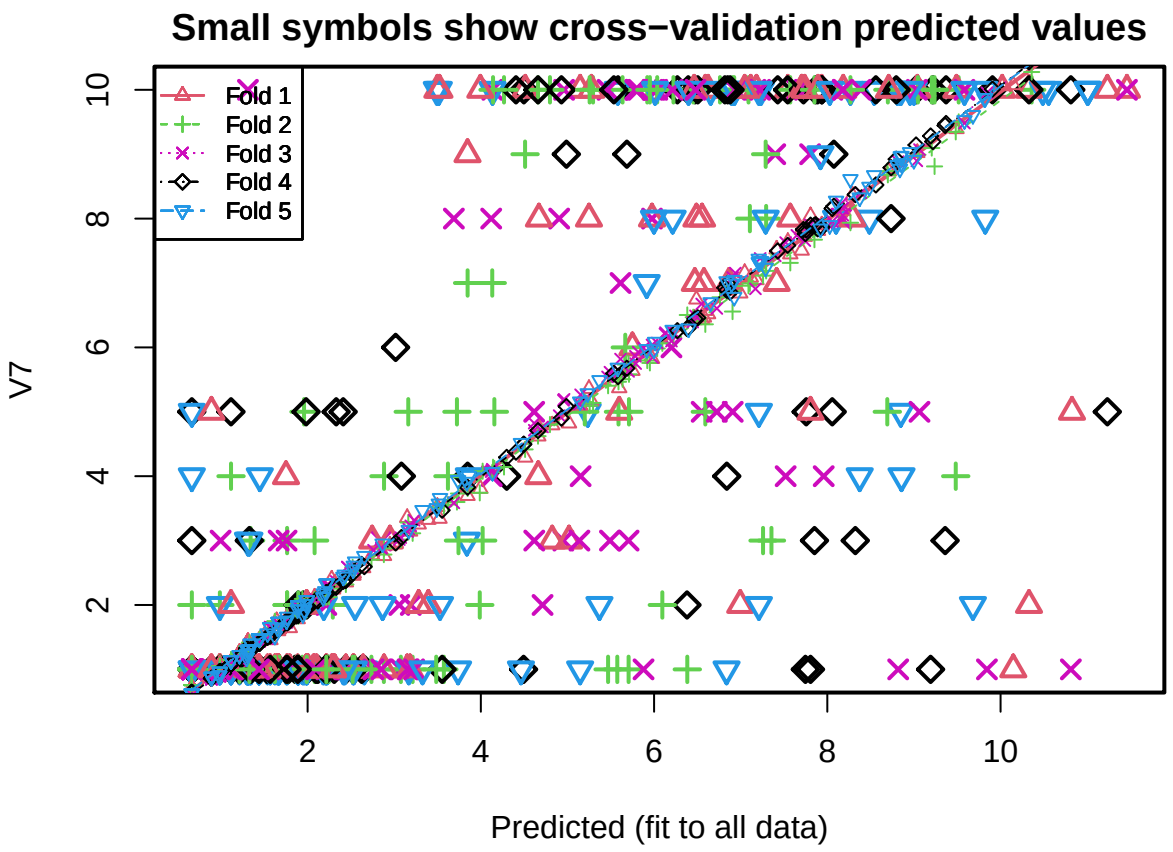
```
## Warning: package 'DAAG' was built under R version 4.0.2
```

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

```
model_cv <- cv.lm(data_modified, model2, m=5)
```

```
## Analysis of Variance Table
##
## Response: V7
##            Df Sum Sq Mean Sq F value  Pr(>F)
## V2          1   3185    3185   616.2 < 2e-16 ***
## V4          1   1683    1683   325.5 < 2e-16 ***
## V5          1    510     510    98.6 < 2e-16 ***
## V8          1    172     172    33.4 1.2e-08 ***
## Residuals 678   3505       5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Warning in cv.lm(data_modified, model2, m = 5):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 136
##                  4       6     12    21    22     29     36    37    51    55
## Predicted    4.663 10.0184  1.213  6.62 6.575  1.213  1.213 10.15  5.02 7.572
## cvpred       4.619 10.0582  1.255  6.52 6.465  1.255  1.255 10.07  4.82 7.447
## V7           4.000 10.0000  1.000 10.00 7.000  1.000  1.000  1.00  3.00 8.000
## CV residual -0.619 -0.0582 -0.255  3.48 0.535 -0.255 -0.255 -9.07 -1.82 0.553
##                 56    57    59    64     67   74     80    81     82   87
## Predicted    5.598 5.750  3.50  3.39  1.990  7.7  1.213  3.15  1.998 4.67
## cvpred       5.369 5.638  3.53  3.32  1.982  7.5  1.255  3.35  1.954 4.75
## V7           5.000 6.000 10.00  2.00  1.000 10.0  1.000  1.00  1.000 8.00
## CV residual -0.369 0.362  6.47 -1.32 -0.982  2.5 -0.255 -2.35 -0.954 3.25
##                 89     91   104     109   118   123    125   127   129   130
## Predicted    1.990  1.311  4.82  0.9873  7.54  7.18  7.415  6.59  4.51 0.663
## cvpred       1.982  1.425  4.79  1.0696  7.64  7.05  7.524  6.57  4.29 0.715
## V7           1.000  1.000  3.00  1.0000 10.00 10.00  7.000 10.00 10.00 1.000
## CV residual -0.982 -0.425 -1.79 -0.0696  2.36  2.95 -0.524  3.43  5.71 0.285
##                131     138    145   156   160    166    169     173    179  187
## Predicted    2.53  1.1158  1.213  5.15  7.94 1.9896  1.763  0.9873  1.990 6.49
## cvpred       2.46  1.0861  1.255  5.10  7.90 1.9819  1.796  1.0696  1.982 6.75
## V7           1.00  1.0000  1.000 10.00 10.00 2.0000  1.000  1.0000  1.000 8.00
## CV residual -1.46 -0.0861 -0.255  4.90  2.10 0.0181 -0.796 -0.0696 -0.982 1.25
##                194    195   197    200   202   210   215    221    226   231
## Predicted    1.311  1.763 6.470  1.440  8.18  2.22 11.46  1.643  0.9873 6.864
## cvpred       1.425  1.796 6.425  1.441  8.12  2.17 11.44  1.752  1.0696 6.883
```

```
## V7           1.000  1.000 7.000  1.000 10.00  1.00 10.00  1.000  1.0000 7.000
## CV residual -0.425 -0.796 0.575 -0.441  1.88 -1.17 -1.44 -0.752 -0.0696 0.117
##                238  248  255    259   272    275    281    302    303    304
## Predicted     6.99 3.85 5.98  1.763  2.22  1.763  1.763  1.311  9.491  1.311
## cvpred        6.84 3.70 5.82  1.796  2.17  1.796  1.796  1.425  9.392  1.425
## V7            2.00 9.00 8.00  1.000  1.00  1.000  1.000  1.000 10.000  1.000
## CV residual -4.84 5.30 2.18 -0.796 -1.17 -0.796 -0.796 -0.425  0.608 -0.425
##                318   324   327  331   341  350   354    358    364   376
## Predicted     8.290  7.12  3.99 6.55  5.28 5.25  7.71  9.039 2.9499 0.663
## cvpred        8.361  7.09  3.81 6.43  5.16 5.36  7.82  9.021 2.9251 0.715
## V7            8.000 10.00 10.00 8.00 10.00 8.00 10.00 10.000 3.0000 1.000
## CV residual -0.361  2.91  6.19 1.57  4.84 2.64  2.18  0.979 0.0749 0.285
##                377     378   381   388    397   403     408   414   417
## Predicted     0.9873  0.9873 0.663  3.18  1.763  2.53  0.9873  2.53  7.74
## cvpred        1.0696  1.0696 0.715  3.14  1.796  2.46  1.0696  2.49  7.72
## V7            1.0000  1.0000 1.000  1.00  1.000  1.00  1.0000  1.00 10.00
## CV residual -0.0696 -0.0696 0.285 -2.14 -0.796 -1.46 -0.0696 -1.49  2.28
##                418   421   426     429   439   447    455    459    473    477
## Predicted     0.9873 2.745 11.23  0.9873  2.64 0.663 0.8897  1.885  1.794  1.659
## cvpred        1.0696 2.774 11.25  1.0696  2.57 0.715 0.9004  1.781  1.643  1.596
## V7            1.0000 3.000 10.00  1.0000  1.00 1.000 1.0000  1.000  1.000  1.000
## CV residual -0.0696 0.226 -1.25 -0.0696 -1.57 0.285 0.0996 -0.781 -0.643 -0.596
##                482   492    498    500   501    502   511   524    528    533
## Predicted     2.88  7.05  1.342  1.666  2.44  1.666 0.663  7.89  1.990  1.311
## cvpred        2.76  7.16  1.272  1.627  2.35  1.627 0.715  7.81  1.982  1.425
## V7            1.00 10.00  1.000  1.000  1.00  1.000 1.000 10.00  1.000  1.000
## CV residual -1.76  2.84 -0.272 -0.627 -1.35 -0.627 0.285  2.19 -0.982 -0.425
##                536   537    539    543    544    551    552   556    563   566
## Predicted     2.28  2.22  1.666  1.568  1.666  1.440  1.311  2.31  1.311 10.33
## cvpred        2.40  2.17  1.627  1.458  1.627  1.441  1.425  2.34  1.425 10.51
## V7            1.00  1.00  1.000  1.000  1.000  1.000  1.000  1.00  1.000 10.00
## CV residual -1.40 -1.17 -0.627 -0.458 -0.627 -0.441 -0.425 -1.34 -0.425 -0.51
##                567   569   570   571    580    590    597   616    617  626  628
## Predicted     2.08  3.52 10.82  6.46  1.311  1.568  1.983  2.30  1.440 1.75 0.89
## cvpred        2.12  3.34 10.79  6.45  1.425  1.458  1.951  2.27  1.441 1.73 0.90
## V7            1.00 10.00  5.00 10.00  1.000  1.000  1.000  1.00  1.000 4.00 5.00
## CV residual -1.12  6.66 -5.79  3.55 -0.425 -0.458 -0.951 -1.27 -0.441 2.27 4.10
##                636   638    641    642   649    663  665   673   676    677
## Predicted     2.07  3.28  2.007  1.440 10.33  1.622  2.1  1.54  2.29  1.305
## cvpred        2.06  3.25  1.926  1.441 10.51  1.717  2.1  1.61  2.24  1.393
## V7            1.00  2.00  1.000  1.000  2.00  1.000  1.0  1.00  1.00  1.000
## CV residual -1.06 -1.25 -0.926 -0.441 -8.51 -0.717 -1.1 -0.61 -1.24 -0.393
##                679   682    688   695    696   699
## Predicted   0.663  8.71  1.440 1.116 0.8897  7.81
## cvpred      0.715  8.74  1.441 1.086 0.9004  8.04
## V7          1.000 10.00  1.000 2.000 1.0000  5.00
## CV residual 0.285  1.26 -0.441 0.914 0.0996 -3.04
##
## Sum of squares = 675    Mean square = 4.96    n = 136
##
## fold 2
## Observations in test set: 137
##                 3    16     17   26    27   40     53   54    62    66    73
## Predicted   1.763  5.58  1.666 3.85  1.44 4.13  5.589 7.11 0.987  3.99  3.57
```

```
## cvpred      1.731  5.54  1.635 3.81  1.44 4.06  5.466 6.99 1.048  3.74  3.53
## V7          2.000  1.00  1.000 7.00  1.00 7.00  5.000 8.00 2.000  2.00  1.00
## CV residual 0.269 -4.54 -0.635 3.19 -0.44 2.94 -0.466 1.01 0.952 -1.74 -2.53
##                77   79    83   85     92     93    99  102    111  112  115
## Predicted    1.94 1.76  2.22 7.29  1.448  1.990 5.667 3.16  2.291 4.51 2.08
## cvpred       2.09 1.73  2.12 7.30  1.478  1.927 5.733 3.28  2.253 4.42 2.08
## V7           1.00 3.00  1.00 9.00  1.000  1.000 6.000 5.00  2.000 9.00 3.00
## CV residual -1.09 1.27 -1.12 1.70 -0.478 -0.927 0.267 1.72 -0.253 4.58 0.92
##                120    139    141    143   150   152  154   172   176   178   182
## Predicted   1.763  1.983  1.116  5.710  7.09  4.27 1.34  1.31  6.76  6.38 0.663
## cvpred      1.731  1.984  1.148  5.582  6.96  4.14 1.34  1.34  6.78  6.50 0.757
## V7          2.000  1.000  1.000  5.000 10.00 10.00 3.00  1.00 10.00  1.00 1.000
## CV residual 0.269 -0.984 -0.148 -0.582  3.04  5.86 1.66 -0.34  3.22 -5.50 0.243
##                183   198   199   204   208    211   218   220   222   225   232
## Predicted    2.44  3.21 0.663  2.22  1.31 10.359  1.31  3.08  6.91  7.57 7.294
## cvpred       2.32  3.11 0.757  2.12  1.34 10.276  1.34  3.02  6.56  7.31 7.188
## V7           1.00  1.00 1.000  1.00  1.00 10.000  1.00  1.00 10.00 10.00 8.000
## CV residual -1.32 -2.11 0.243 -1.12 -0.34 -0.276 -0.34 -2.02  3.44  2.69 0.812
##                233   235   237   242    243  244   254    261   265   267   268
## Predicted    5.47  2.08  6.22  2.43  1.537 1.96  6.92  9.206  7.26  5.92  4.66
## cvpred       5.40  2.08  6.16  2.39  1.535 1.92  6.95  9.096  7.11  5.80  4.62
## V7           1.00  1.00 10.00  1.00  1.000 5.00 10.00 10.000  3.00 10.00 10.00
## CV residual -4.40 -1.08  3.84 -1.39 -0.535 3.08  3.05  0.904 -4.11  4.20  5.38
##                271  274   279    282    287   288  289   290   291   292   294
## Predicted    4.8 3.62  1.31  1.870  9.516  1.44 3.72  6.45 0.663  1.31  5.64
## cvpred       4.8 3.56  1.34  1.865  9.502  1.44 3.60  6.47 0.757  1.34  5.51
## V7          10.0 4.00  1.00  1.000 10.000  1.00 5.00 10.00 1.000  1.00 10.00
## CV residual  5.2 0.44 -0.34 -0.865  0.498 -0.44 1.40  3.53 0.243 -0.34  4.49
##                297   299   306   312   317   319    323    328   333   344   349
## Predicted   4.16  2.25  5.30 0.663  4.14  1.31  1.763  0.9873  2.54 0.663  5.71
## cvpred      4.12  2.13  5.14 0.757  4.14  1.34  1.731  1.0481  2.51 0.757  5.75
## V7          5.00  1.00 10.00 1.000 10.00  1.00  1.000  1.0000  1.00 1.000  1.00
## CV residual 0.88 -1.13  4.86 0.243  5.86 -0.34 -0.731 -0.0481 -1.51 0.243 -4.75
##                351   353   362    366   368    371   374    395    398  404
## Predicted    2.23  4.02  6.04  1.213  9.05  1.983  2.22  1.622  1.342 1.12
## cvpred       2.20  4.08  5.98  1.244  8.99  1.984  2.16  1.745  1.344 1.15
## V7           1.00  3.00 10.00  1.000 10.00  1.000  1.00  1.000  1.000 4.00
## CV residual -1.20 -1.08  4.02 -0.244  1.01 -0.984 -1.16 -0.745 -0.344 2.85
##                413   416   428   432    433   441  442   444   450   452   454
## Predicted    9.48  3.74  6.10  2.88  1.892  9.24 2.88 0.663  8.70  1.57  7.85
## cvpred       9.35  3.73  6.06  2.78  1.831  8.81 2.88 0.757  8.68  1.54  7.67
## V7           4.00  3.00  2.00  1.00  1.000 10.00 4.00 2.000 10.00  1.00 10.00
## CV residual -5.35 -0.73 -4.06 -1.78 -0.831  1.19 1.12 1.243  1.32 -0.54  2.33
##                471   475   481    495   505     518    519   525   531   541
## Predicted    1.44  1.57  1.57  5.200 0.663  0.9873  1.765  1.44  5.26 1.892
## cvpred       1.44  1.54  1.54  5.151 0.757  1.0481  1.827  1.44  5.09 1.831
## V7           1.00  1.00  1.00  5.000 1.000  1.0000  1.000  1.00 10.00 2.000
## CV residual -0.44 -0.54 -0.54 -0.151 0.243 -0.0481 -0.827 -0.44  4.91 0.169
##                549   550   553   557   558    560   564    574    577    579
## Predicted    1.116  6.59  2.74  2.54  2.23  1.892  1.44  0.9873  1.892  0.9873
## cvpred       1.148  6.36  2.70  2.51  2.20  1.831  1.44  1.0481  1.831  1.0481
## V7           1.000  5.00  1.00  1.00  1.00  1.000  1.00  1.0000  1.000  1.0000
## CV residual -0.148 -1.36 -1.70 -1.51 -1.20 -0.831 -0.44 -0.0481 -0.831 -0.0481
##                593   600   601    607   611    615   624   633   640   644   658
```

```
## Predicted    5.95  2.52  1.44  1.674  8.27  1.213 0.663 0.663  2.23 0.663  3.48
## cvpred       5.76  2.59  1.44  1.674  7.94  1.244 0.757 0.757  2.20 0.757  3.52
## V7          10.00  1.00  1.00  1.000 10.00  1.000 1.000 1.000  1.00 1.000  1.00
## CV residual  4.24 -1.59 -0.44 -0.674  2.06 -0.244 0.243 0.243 -1.20 0.243 -2.52
##                662    664   666   670   683   685    691   697
## Predicted    1.990  1.622 0.663  8.69  2.22 0.663  1.328  7.35
## cvpred       1.927  1.745 0.757  8.74  2.12 0.757  1.417  7.38
## V7           1.000  1.000 1.000  5.00  1.00 1.000  1.000  3.00
## CV residual -0.927 -0.745 0.243 -3.74 -1.12 0.243 -0.417 -4.38
##
## Sum of squares = 672    Mean square = 4.9    n = 137
##
## fold 3
## Observations in test set: 137
##                 7     10     11  15     23     30    32    42    45    46   50
## Predicted    1.31  1.666  1.311 7.8  1.440  1.298  1.54  4.95  8.82 0.987 4.90
## cvpred       1.21  1.702  1.211 7.9  1.433  1.172  1.48  5.18  8.89 0.894 4.94
## V7          10.00  1.000  1.000 9.0  1.000  1.000  1.00  3.00  1.00 1.000 8.00
## CV residual  8.79 -0.702 -0.211 1.1 -0.433 -0.172 -0.48 -2.18 -7.89 0.106 3.06
##                61   63    65   69    71    78    84      86    88    94     97
## Predicted    5.14 5.98 0.987 7.40  2.53  2.22  3.06  4.1260  5.98 0.987  1.222
## cvpred       5.12 6.04 0.894 7.42  2.57  2.30  3.02  4.0896  5.84 0.894  1.179
## V7           3.00 8.00 1.000 9.00  1.00  1.00  2.00  4.0000 10.00 1.000  1.000
## CV residual -2.12 1.96 0.106 1.58 -1.57 -1.30 -1.02 -0.0896  4.16 0.106 -0.179
##                101   105   106   108   124    136  147   149    151   162   164
## Predicted   4.616 10.81  4.62  7.17  4.13  2.216 3.69  3.08  1.311  1.99 0.996
## cvpred      4.823 10.88  4.71  6.91  4.11  2.288 3.59  3.05  1.211  2.02 0.910
## V7          5.000  1.00  3.00 10.00 10.00  2.000 8.00  1.00  1.000  1.00 3.000
## CV residual 0.177 -9.88 -1.71  3.09  5.89 -0.288 4.41 -2.05 -0.211 -1.02 2.090
##                167   175   184   185   186   201    203   207   219   240   241
## Predicted    6.45  6.13  8.06  6.13  1.54  7.65  1.311  6.55  7.96  4.96  3.19
## cvpred       6.39  6.23  7.96  6.21  1.48  7.74  1.211  6.67  7.91  5.19  3.23
## V7          10.00 10.00 10.00 10.00  1.00 10.00  1.000  5.00  4.00 10.00  2.00
## CV residual  3.61  3.77  2.04  3.79 -0.48  2.26 -0.211 -1.67 -3.91  4.81 -1.23
##                245   249  260   269    270   296    307   315    325   330
## Predicted    1.311  1.99 4.12  7.52  1.311  7.74  1.311 0.987  1.311  7.22
## cvpred       1.211  2.02 4.07  7.63  1.211  7.66  1.211 0.894  1.211  7.38
## V7           1.000  1.00 8.00  4.00  1.000 10.00  1.000 1.000  1.000 10.00
## CV residual -0.211 -1.02 3.93 -3.63 -0.211  2.34 -0.211 0.106 -0.211  2.62
##                334   335   337   340   346   357   359  360  363    367    369
## Predicted    5.79  5.46  6.04  5.82 0.663 2.850  5.15 5.61 1.76  9.583  1.298
## cvpred       5.75  5.54  6.07  5.93 0.578 2.882  5.26 5.82 1.73  9.483  1.172
## V7          10.00 10.00 10.00 10.00 1.000 3.000  4.00 7.00 3.00 10.000  1.000
## CV residual  4.25  4.46  3.93  4.07 0.422 0.118 -1.26 1.18 1.27  0.517 -0.172
##                373   375   380   382   384   385   387    389    391   415
## Predicted    1.983  1.76  3.17  6.94 0.890 0.890  6.21  1.213  1.320  5.78
## cvpred       1.999  1.73  3.18  7.15 0.847 0.847  6.24  1.164  1.226  5.86
## V7           1.000  1.00  1.00 10.00 1.000 1.000 10.00  1.000  1.000 10.00
## CV residual -0.999 -0.73 -2.18  2.85 0.153 0.153  3.76 -0.164 -0.226  4.14
##                427   437   446    448   449   451   461   463    464   472
## Predicted    3.15  5.88 0.890  1.568 0.663  2.33  2.23  2.46  1.342  2.46
## cvpred       3.14  5.99 0.847  1.655 0.578  2.37  2.32  2.59  1.386  2.59
## V7           1.00  1.00 1.000  1.000 1.000  1.00  1.00  1.00  1.000  1.00
## CV residual -2.14 -4.99 0.153 -0.655 0.422 -1.37 -1.32 -1.59 -0.386 -1.59
```

```
##                479   480   484    487   489   491   494   497   507    512
## Predicted    1.568  8.18  8.39  1.440  5.71 0.663  9.03 0.663  9.06  1.892
## cvpred       1.655  8.05  8.40  1.433  5.63 0.578  8.90 0.578  9.07  1.972
## V7           1.000 10.00 10.00  1.000  3.00 1.000 10.00 1.000  5.00  1.000
## CV residual -0.655  1.95  1.60 -0.433 -2.63 0.422  1.10 0.422 -4.07 -0.972
##                513    514   521    522   523   529    530    534    542    546
## Predicted    1.568  1.440 0.663  1.342  6.91  2.76  1.666  1.440  1.116  1.892
## cvpred       1.655  1.433 0.578  1.386  7.08  2.85  1.702  1.433  1.117  1.972
## V7           1.000  1.000 1.000  1.000  5.00  1.00  1.000  1.000  1.000  1.000
## CV residual -0.655 -0.433 0.422 -0.386 -2.08 -1.85 -0.702 -0.433 -0.117 -0.972
##                548    555   561  568   585    588   598    603   605    606
## Predicted    0.890  1.116  2.22 1.67  3.23  1.892  2.85  1.666  6.48  8.162
## cvpred       0.847  1.117  2.29 1.70  3.31  1.972  2.88  1.702  6.49  8.237
## V7           1.000  1.000  1.00 3.00  1.00  1.000  1.00  1.000 10.00  8.000
## CV residual 0.153 -0.117 -1.29 1.30 -2.31 -0.972 -1.88 -0.702  3.51 -0.237
##                614   622     627    630   634   637    639    650    655    656
## Predicted    1.213  4.71  6.2002  1.342  5.49  9.84  1.342  1.440  1.763  1.440
## cvpred       1.164  4.76  6.0967  1.386  5.62  9.95  1.386  1.433  1.749  1.433
## V7           1.000  2.00  6.0000  1.000  3.00  1.00  1.000  1.000  1.000  1.000
## CV residual -0.164 -2.76 -0.0967 -0.386 -2.62 -8.95 -0.386 -0.433 -0.749 -0.433
##                659   660   661   672   675   681   684   686   690   692    693
## Predicted    8.18 0.663 0.987  2.10 0.987 11.46 0.663 0.663 0.663  6.72  1.116
## cvpred       8.16 0.578 0.894  2.08 0.894 11.51 0.578 0.578 0.578  6.60  1.117
## V7          10.00 1.000 1.000  1.00 1.000 10.00 1.000 1.000 1.000  5.00  1.000
## CV residual  1.84 0.422 0.106 -1.08 0.106 -1.51 0.422 0.422 0.422 -1.60 -0.117
##                694
## Predicted    1.440
## cvpred       1.433
## V7           1.000
## CV residual -0.433
##
## Sum of squares = 835    Mean square = 6.09    n = 137
##
## fold 4
## Observations in test set: 137
##                2     5      8    20    34     35    39   47    48    52    58
## Predicted    4.5  2.65  1.855  2.44  1.87  1.757  6.47 4.99 0.987 3.847  4.49
## cvpred       4.5  2.60  1.847  2.38  1.83  1.746  6.44 5.08 0.963 3.827  4.50
## V7          10.0  1.00  1.000  1.00  1.00  1.000 10.00 9.00 1.000 4.000  1.00
## CV residual  5.5 -1.60 -0.847 -1.38 -0.83 -0.746  3.56 3.92 0.037 0.173 -3.50
##                 70    72     75     76     90     95   103  110   116   119
## Predicted    1.311  6.38  4.298 1.9521  1.546  1.537  2.31 5.69 0.663 0.663
## cvpred       1.284  6.29  4.292 1.9488  1.509  1.504  2.29 5.68 0.642 0.642
## V7           1.000  2.00  4.000 2.0000  1.000  1.000  1.00 9.00 5.000 3.000
## CV residual -0.284 -4.29 -0.292 0.0512 -0.509 -0.504 -1.29 3.32 4.358 2.358
##                122    132   133   155    163   168    170     171    177    181
## Predicted   1.9896  1.537  7.43 0.663  1.763  9.19 0.9958  1.1158  1.537  1.311
## cvpred      1.9439  1.504  7.50 0.642  1.724  9.28 0.9675  1.0814  1.504  1.284
## V7          2.0000  1.000 10.00 1.000  1.000  1.00 1.0000  1.0000  1.000  1.000
## CV residual 0.0561 -0.504  2.50 0.358 -0.724 -8.28 0.0325 -0.0814 -0.504 -0.284
##                188    193    206    212    213   216   217  223   227   228
## Predicted    9.219  1.892  9.025  8.736  1.311  7.76 0.987 2.33  7.87  8.06
## cvpred       9.196  1.842  9.144  8.791  1.284  7.82 0.963 2.27  7.91  8.16
## V7          10.000  1.000 10.000  8.000  1.000  5.00 1.000 5.00 10.00  5.00
```

```
## CV residual  0.804 -0.842  0.856 -0.791 -0.284 -2.82 0.037 2.73  2.09 -3.16
##               234   239    247   252   253   263   264   273   283   300   305
## Predicted    6.27 8.076  9.371  7.94  4.41  7.72  7.94  4.66  5.58  6.39  6.50
## cvpred       6.26 8.192  9.477  7.87  4.37  7.83  7.87  4.71  5.55  6.30  6.46
## V7          10.00 9.000 10.000 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
## CV residual  3.74 0.808  0.523  2.13  5.63  2.17  2.13  5.29  4.45  3.70  3.54
##               309  310    311   321   332   336    338   339    342   343   345
## Predicted    7.85 2.41  1.213  4.93  2.22 0.663  1.311 0.987  1.311 0.890  7.89
## cvpred       7.90 2.37  1.183  4.90  2.16 0.642  1.284 0.963  1.284 0.862  7.84
## V7           3.00 5.00  1.000 10.00  1.00 1.000  1.000 1.000  1.000 1.000 10.00
## CV residual -4.90 2.63 -0.183  5.10 -1.16 0.358 -0.284 0.037 -0.284 0.138  2.16
##               347   348    352   355    356     361    365    372   379   386
## Predicted    2.22 0.663  1.537 0.987  1.666  9.9068  1.537  1.298  2.44  1.77
## cvpred       2.19 0.642  1.504 0.963  1.623 10.0105  1.504  1.328  2.38  1.75
## V7           1.00 1.000  1.000 1.000  1.000 10.0000  1.000  1.000  1.00  1.00
## CV residual -1.19 0.358 -0.504 0.037 -0.623 -0.0105 -0.504 -0.328 -1.38 -0.75
##               390   392    393   394     402    405   406    407    410   411
## Predicted   1.892  7.54  1.440 0.663  1.1158  1.328 0.987  1.983  1.757 0.987
## cvpred      1.842  7.58  1.403 0.642  1.0814  1.293 0.963  1.966  1.746 0.963
## V7          2.000 10.00  1.000 1.000  1.0000  1.000 1.000  1.000  1.000 1.000
## CV residual 0.158  2.42 -0.403 0.358 -0.0814 -0.293 0.037 -0.966 -0.746 0.037
##               420   423   431   434   436    438  443   445    453  456   457
## Predicted   0.890  2.62 0.987  2.10  6.85  1.342 1.33  3.55  1.780 3.02  8.56
## cvpred      0.862  2.63 0.963  2.08  6.99  1.301 1.29  3.47  1.733 2.96  8.52
## V7          1.000  1.00 1.000  1.00 10.00  1.000 3.00  1.00  1.000 6.00 10.00
## CV residual 0.138 -1.63 0.037 -1.08  3.01 -0.301 1.71 -2.47 -0.733 3.04  1.48
##               458  462    465    474   483    485   488   490    496    503
## Predicted    9.36 1.12  1.342  1.342 11.23  1.885 10.81 3.083  1.440  1.998
## cvpred       9.45 1.08  1.301  1.301 11.31  1.864 10.89 3.048  1.403  1.948
## V7           3.00 5.00  1.000  1.000  5.00  1.000 10.00 4.000  1.000  1.000
## CV residual -6.45 3.92 -0.301 -0.301 -6.31 -0.864 -0.89 0.952 -0.403 -0.948
##                504     506    509   516   520    526    527    535   538   540
## Predicted    1.990  1.1158  1.568  6.88  6.82  1.448  1.342  1.213  2.53  2.12
## cvpred       1.944  1.0814  1.521  6.86  6.93  1.407  1.301  1.183  2.51  2.06
## V7           1.000  1.0000  1.000 10.00 10.00  1.000  1.000  1.000  1.00  1.00
## CV residual -0.944 -0.0814 -0.521  3.14  3.07 -0.407 -0.301 -0.183 -1.51 -1.06
##               554    559   562   572    573   578   581     584   586   589
## Predicted   1.98  1.213  2.22  8.80  1.440 0.987  2.21  1.1158 0.663  8.32
## cvpred      1.97  1.183  2.16  8.92  1.403 0.963  2.19  1.0814 0.642  8.37
## V7          5.00  1.000  1.00 10.00  1.000 1.000  1.00  1.0000 1.000  3.00
## CV residual 3.03 -0.183 -1.16  1.08 -0.403 0.037 -1.19 -0.0814 0.358 -5.37
##               591   595    599   604    609    632    646   667    668    674
## Predicted    7.81  5.54  1.440  7.75 10.328  1.892  1.440  2.22  1.763  1.885
## cvpred       7.90  5.60  1.403  7.77 10.433  1.842  1.403  2.19  1.724  1.864
## V7           1.00 10.00  1.000  1.00 10.000  1.000  1.000  1.00  1.000  1.000
## CV residual -6.90  4.40 -0.403 -6.77 -0.433 -0.842 -0.403 -1.19 -0.724 -0.864
##                678    689   698
## Predicted    1.568  1.342  6.84
## cvpred       1.521  1.301  6.89
## V7           1.000  1.000  4.00
## CV residual -0.521 -0.301 -2.89
##
## Sum of squares = 779    Mean square = 5.68    n = 137
##
```

```
## fold 5
## Observations in test set: 136
##                  1     9    13   14    18    19    25     28     31    33    38
## Predicted     2.22 0.890  3.84 1.31  1.99  7.24  1.31  1.892  1.440  7.21  3.74
## cvpred        2.33 0.883  3.91 1.30  2.07  7.34  1.30  1.993  1.478  7.32  3.94
## V7            1.00 1.000  3.00 3.00  1.00 10.00  1.00  1.000  1.000  5.00  1.00
## CV residual -1.33 0.117 -0.91 1.70 -1.07  2.66 -0.30 -0.993 -0.478 -2.32 -2.94
##                 43    44    49    60    68    96    98   100   107   113    114
## Predicted     6.92  5.15  2.65  5.37  3.49  1.31  2.22  8.54  8.85  8.27  8.486
## cvpred        6.78  5.16  2.76  5.49  3.50  1.30  2.33  8.67  8.86  8.61  8.492
## V7           10.00  1.00  1.00  2.00 10.00  1.00  1.00 10.00 10.00 10.00  8.000
## CV residual  3.22 -4.16 -1.76 -3.49  6.50 -0.30 -1.33  1.33  1.14  1.39 -0.492
##                117    121    126    128    134    135    137   142   144   148
## Predicted     3.53  1.961 0.9873  1.763  1.440  1.440  1.666 0.890 0.663 0.987
## cvpred        3.66  1.921 0.9625  1.815  1.478  1.478  1.735 0.883 0.625 0.962
## V7            2.00  1.000 1.0000  1.000  1.000  1.000  1.000 1.000 5.000 2.000
## CV residual -1.66 -0.921 0.0375 -0.815 -0.478 -0.478 -0.735 0.117 4.375 1.038
##                153   157    158   161    174   180    189    190   191   192
## Predicted     8.85  1.30  1.537  6.89 10.554  3.51  8.101  1.946  9.82  8.84
## cvpred        8.97  1.24  1.557  6.99 10.538  3.57  8.276  1.856  9.87  8.77
## V7            5.00  1.00  1.000 10.00 10.000 10.00  8.000  1.000  8.00 10.00
## CV residual -3.97 -0.24 -0.557  3.01 -0.538  6.43 -0.276 -0.856 -1.87  1.23
##                196   205   209    214  224   229   230    246   251   256   257
## Predicted     1.99  1.31  1.31 10.488 6.21  1.31  8.81  2.548 0.981  4.13  1.12
## cvpred        2.07  1.30  1.30 10.557 6.27  1.30  8.83  2.673 0.903  4.06  1.14
## V7            1.00  1.00  1.00 10.000 8.00  1.00 10.00  2.000 1.000 10.00  1.00
## CV residual -1.07 -0.30 -0.30 -0.557 1.73 -0.30  1.17 -0.673 0.097  5.94 -0.14
##                258   262   266    277    278  280   284   285   286   301   308
## Predicted     1.440  9.00  3.17  1.440 0.9873 5.91  5.58  6.93 11.01  8.37  1.31
## cvpred        1.478  8.91  3.16  1.478 0.9625 5.97  5.68  7.04 11.05  8.33  1.30
## V7            1.000 10.00  1.00  1.000 1.0000 7.00 10.00 10.00 10.00  4.00  1.00
## CV residual -0.478  1.09 -2.16 -0.478 0.0375 1.03  4.32  2.96 -1.05 -4.33 -0.30
##                313   314    320    326      329    370   383    396    399
## Predicted     6.84 0.663  5.233  1.757  3.86109  1.622  2.41  1.440  1.440
## cvpred        7.02 0.625  5.285  1.756  4.00182  1.518  2.44  1.478  1.478
## V7            1.00 1.000  5.000  1.000  4.00000  1.000  1.00  1.000  1.000
## CV residual -6.02 0.375 -0.285 -0.756 -0.00182 -0.518 -1.44 -0.478 -0.478
##                400  401    409    419    422   424   425   430  435    440
## Predicted     1.298 7.92  2.187  2.865  9.815  2.53  1.12  1.21 6.00  1.568
## cvpred        1.181 7.84  2.178  2.951  9.865  2.55  1.14  1.22 5.96  1.655
## V7            1.000 9.00  2.000  2.000 10.000  1.00  1.00  1.00 8.00  1.000
## CV residual -0.181 1.16 -0.178 -0.951  0.135 -1.55 -0.14 -0.22 2.04 -0.655
##                460   466   467   468    469   470   476    478  486    493
## Predicted     2.20  8.86  7.21  6.65  1.342  1.30  1.12  1.342 1.33  1.666
## cvpred        2.21  8.97  7.38  6.70  1.398  1.24  1.14  1.398 1.31  1.735
## V7            1.00  4.00 10.00 10.00  1.000  1.00  1.00  1.000 3.00  1.000
## CV residual -1.21 -4.97  2.62  3.30 -0.398 -0.24 -0.14 -0.398 1.69 -0.735
##                499   508   510   515   517   532   545    547   565   575   576
## Predicted     1.666 0.663 0.890  8.95 0.663  1.98  2.18  9.583  1.99  7.21  2.53
## cvpred        1.735 0.625 0.883  9.03 0.625  2.01  2.12  9.526  2.07  7.32  2.61
## V7            1.000 4.000 1.000 10.00 1.000  1.00  1.00 10.000  1.00  2.00  1.00
## CV residual -0.735 3.375 0.117  0.97 0.375 -1.01 -1.12  0.474 -1.07 -5.32 -1.61
##                582   583   587   592    594    596    602   608    610   612
## Predicted     8.03  6.01 11.01  6.40  1.885  1.892 0.9873 0.663  1.568  9.68
```

```
## cvpred       7.90  6.08 11.05  6.29  1.933  1.993 0.9625 0.625  1.655  9.61
## V7          10.00 10.00 10.00 10.00  1.000  1.000 1.0000 1.000  1.000  2.00
## CV residual  2.10  3.92 -1.05  3.71 -0.933 -0.993 0.0375 0.375 -0.655 -7.61
##                613    619    620    621   623   625   629   631   635    643
## Predicted   11.01  1.666  1.892  1.440  3.33  2.22 0.890  2.43  1.12  1.440
## cvpred      11.05  1.735  1.993  1.478  3.47  2.34 0.883  2.47  1.14  1.478
## V7          10.00  1.000  1.000  1.000  1.00  1.00 1.000  1.00  1.00  1.000
## CV residual -1.05 -0.735 -0.993 -0.478 -2.47 -1.34 0.117 -1.47 -0.14 -0.478
##                645   647    648  651    652    653    654    657   669   671
## Predicted   0.890 0.981  1.328 1.45  1.652  1.892  1.666  1.892  4.46 7.288
## cvpred      0.883 0.903  1.311 1.48  1.649  1.993  1.735  1.993  4.51 7.234
## V7          1.000 1.000  1.000 4.00  1.000  1.000  1.000  1.000  1.00 8.000
## CV residual 0.117 0.097 -0.311 2.52 -0.649 -0.993 -0.735 -0.993 -3.51 0.766
##                680   687
## Predicted   0.890 0.663
## cvpred      0.883 0.625
## V7          1.000 1.000
## CV residual 0.117 0.375
##
## Sum of squares = 591    Mean square = 4.35    n = 136
##
## Overall (Sum over all 136 folds)
##  ms
## 5.2
```

```
SST <- sum((as.numeric(data[-missing,]$V7) - mean(as.numeric(data[-missing,]$V7)))^2)
R2_cv <- 1 - attr(model_cv,"ms")*nrow(data[-missing,])/SST

# predict missing V7 values.
V7_hat <- predict(model2, newdata = data[missing,])

# Impute  missing V7 with predicted values of linear model.
data_reg_imp <- data
data_reg_imp[missing,]$V7 <- V7_hat
data_reg_imp$V7 <- as.numeric(data_reg_imp$V7)

# Round the V7_hat values, originals are all integer
data_reg_imp[missing,]$V7 <- round(V7_hat)
data_reg_imp$V7 <- as.integer(data_reg_imp$V7)

# Make sure no V7 values are outside of the original range.
data_reg_imp$V7[data_reg_imp$V7 > 10] <- 10
data_reg_imp$V7[data_reg_imp$V7 < 1] <- 1
```

## Regression with Perturbation Imputation

```
set.seed(1)
# Perturb the predictions for missing V7 values with a random normal distriubtion
V7_hat_pert <- rnorm(nrow(data[missing,]), V7_hat, sd(V7_hat))
V7_hat_pert
```

```
##  [1]  4.078  8.386 -0.855  5.138  1.707  0.407  3.790  3.391  3.343  5.413
## [11]  4.320  3.386  3.875 -3.118  3.467  0.564
```

```
data_reg_pert_imp <- data
data_reg_pert_imp[missing,]$V7 <- V7_hat_pert
data_reg_pert_imp$V7 <- as.numeric(data_reg_pert_imp$V7)

# Round the V7_hat_pert values to integers.

data_reg_pert_imp[missing,]$V7 <- round(V7_hat_pert)
(data_reg_pert_imp$V7 <- as.integer(data_reg_pert_imp$V7))
```

```
##   [1]  1 10  2  4  1 10 10  1  1  1  1  1  3  3  9  1  1  1 10  1 10  7  1  4  1
##  [26]  7  1  1  1  1  1  1  5  1  1  1  1  1 10  7  8  3 10  1  1  1  9  1  1  8
##  [51]  3  4  5  8  8  5  6  1 10  2  3  2  8  2  1  2  1 10  9  1  1  2  1 10  4
##  [76]  2  1  1  3  1  1  1  1  2  9  4  8 10  1  1  1  1  1  1  1  1  1  1  6 10
## [101]  5  5  1  3  1  3 10 10  1  9  2  9 10  8  3  5  2 10  3  2  1  2 10 10  7
## [126]  1 10  1 10  1  1  1 10  1  1  2  1  1  1 -1  1  1  5  5  1  5  8  2  1 10
## [151]  1 10  5  3  1 10  1  1  2 10 10  1  1  3  0  2 10  1  1  1  1  1  1 10 10
## [176] 10  1  1  1 10  1  1  1 10 10  1  8 10  8  1  8 10  1  1  1  1  7  1  1  1
## [201] 10 10  1  1  1 10  5  1  1  1 10  8  1 10 10  5  1  1  4  1  1 10  5  8 10
## [226]  1 10  5  1 10  7  8  1 10  1  4 10  2  9 10  2  1  1  5  1  2 10  9  1  3
## [251]  1 10 10 10  8 10  1  1  1  8 10 10 10 10  3  1 10 10  4  1 10  1 10  4  1
## [276]  3  1  1  1  7  1  1 10 10 10 10 10  1  5 10  1  1  5 10  4 10  5  3  1 10
## [301]  4  1 10  1 10 10  1  1  3  5  1  1  1  1  1  4 10  8  1  5 10 -3  1 10  1
## [326]  1 10  1  4 10  8  1  1 10 10  1 10  1  1 10 10  1  1  1 10  1  1  1  1  8
## [351]  1  1  3 10  1  1  3 10  4  7 10 10  3  3  1  1 10 10  1  1  1  1  1  1  1
## [376]  1  1  1  1  1  1 10  1  1  1  1 10  1  1  2  1 10  1  1  1  1  1  1  1  1
## [401]  9  1  1  4  1  1  1  1  2  1  1  3  4  1 10  3 10  1  2  1  3 10  1  1  1
## [426] 10  1  2  1  1  1  1  1  1  8 10  1  1  1  1 10  4  3  2  1  1  1  1  1 10
## [451]  1  1  1 10  1  6 10  3  1  1  1  5  1  1  1  4 10 10  1  1  1  1  1  1  1
## [476]  1  1  1  1 10  1  1  5 10  1  3  1 10  3  4  1 10  1 10  5  1  1  1  1  1
## [501]  1  1  1  1  1  1  5  4  1  1  1  1  1  1 10 10  1  1  1 10  1  1  5 10  1
## [526]  1  1  1  1  1 10  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1 10  1  1  5
## [551]  1  1  1  5  1  1  1  1  1  1  1  1  1  1  1 10  1  3 10  5 10 10  1  1  2
## [576]  1  1  1  1  1  1 10 10  1  1  1 10  1  3  1  1 10 10  1 10  1  1  1  1  1
## [601]  1  1  1  1 10  8  1  1 10  1 10  2 10  1  1  1  1  1  1  1  1  2  1  1  1
## [626]  4  6  5  1  1  1  1  1  3  1  1  1  2  1  1  1  1  1  1  1  1  1  1  2  1
## [651]  4  1  1  1  1  1  1  1 10  1  1  1  1  1  1  1  1  1  1  5  8  1  1  1  1
## [676]  1  1  1  1  1 10 10  1  1  1  1  1  1  1  1  1  5  1  1  2  1  3  4  5
```

```
# Make sure no V7 values are outside of the orignal range.

data_reg_pert_imp$V7[data_reg_pert_imp$V7 > 10] <- 10
data_reg_pert_imp$V7[data_reg_pert_imp$V7 < 1] <- 1
```

## Classifcation Models

```
set.seed(1)
# split data
```

```r
training <- sample(nrow(data), size = floor(nrow(data) * 0.7))
validation <- setdiff(1:nrow(data), training)
#  KNN Models
library(kknn)
```

```
## Warning: package 'kknn' was built under R version 4.0.2
```

```r
acc_knn <- rep(0,25)

for (k in 1:5) {

  knn_model <- kknn(V11~V2+V3+V4+V5+V6+V7+V8+V9+V10, data_mode_imp[training,], data_mode_imp[validation

  # Compare models using validation set.

  pred <- as.integer(fitted(knn_model)+0.5) # round off to 2 or 4

  acc_knn[k] = sum(pred == data_mode_imp[validation,]$V11) / nrow(data_mode_imp[validation,])
}

# Data with regression imputation

for (k in 1:5) {

  # Fit k-nearest-neighbor model using training set, validate on test set.

  knn_model <- kknn(V11~V2+V3+V4+V5+V6+V7+V8+V9+V10, data_reg_imp[training,], data_reg_imp[validation,]

  # Compare models using validation set.

  pred <- as.integer(fitted(knn_model)+0.5) # round off to 2 or 4

  acc_knn[k+5] = sum(pred == data_reg_imp[validation,]$V11) / nrow(data_reg_imp[validation,])
}

# Data with regression with perturbation imputation

for (k in 1:5) {

  # Fit k-nearest-neighbor model using training set, validate on test set.

  knn_model <- kknn(V11~V2+V3+V4+V5+V6+V7+V8+V9+V10, data_reg_pert_imp[training,], data_reg_pert_imp[va

  # Compare models using validation set.

  pred <- as.integer(fitted(knn_model)+0.5) # round off to 2 or 4

  acc_knn[k+10] = sum(pred == data_reg_pert_imp[validation,]$V11) / nrow(data_reg_pert_imp[validation,]]
}

# Check how many of the missing indices fall into the training set

length(intersect(missing, training))/length(missing)
```

```
## [1] 0.625
```

```
# Since this is relatively close to 70% and since the number of observations with missing values is so

training_no_missing <- setdiff(training, intersect(missing, training))
validation_no_missing <- setdiff(validation, intersect(missing, validation))

# Replace missing data with 0's so that V7 can be read as type integer and skip over missing values in

data_no_missing <- data
data_no_missing$V7[data$V7 == "?"] <- 0
data_no_missing$V7 <- as.integer(data_no_missing$V7)

for (k in 1:5) {

  # Fit k-nearest-neighbor model using training set, validate on test set.

  knn_model <- kknn(V11~V2+V3+V4+V5+V6+V7+V8+V9+V10, data_no_missing[training_no_missing,],
                    data_no_missing[validation_no_missing,], k=k)

  # Compare models using validation set.

  pred <- as.integer(fitted(knn_model)+0.5) # round off to 2 or 4

  acc_knn[k+15] = sum(pred == data_no_missing[validation_no_missing,]$V11) /
    nrow(data_no_missing[validation_no_missing,])
}

# Add a binary variable to the original data to indicate if an observation has a missing V7 value.

data_binary <- data
data_binary$V12[data$V7 == "?"] <- 0
data_binary$V12[data$V7 != "?"] <- 1

# Create interaction factor for V7 and V12.

data_binary$V13[data$V7 == "?"] <- 0
data_binary$V13[data$V7 != "?"] <- as.integer(data[-missing,]$V7)

# Use the interaction factor in the modeling.

for (k in 1:5) {

  # Fit k-nearest-neighbor model using training set, validate on test set.

  knn_model <- kknn(V11~V2+V3+V4+V5+V6+V8+V9+V10+V13, data_binary[training,], data_binary[validation,],

  # Compare models using validation set.

  pred <- as.integer(fitted(knn_model)+0.5) # round off to 2 or 4

  acc_knn[k+20] = sum(pred == data_binary[validation,]$V11) / nrow(data_binary[validation,])
}
```

```
acc_knn
```

```
##  [1] 0.943 0.943 0.900 0.900 0.900 0.948 0.948 0.905 0.905 0.905 0.948 0.948
## [13] 0.900 0.900 0.900 0.951 0.951 0.912 0.912 0.912 0.948 0.948 0.900 0.900
## [25] 0.900
```

```
# SVM Models
library(kernlab)
acc_svm<- rep(0,30)

amounts <- c(0.0001, 0.001, 0.01, 0.1, 1, 10)

# Data with mode imputation

for (i in 1:6) {

  # Fit model using training set.

  model_svm <- ksvm(as.matrix(data_mode_imp[training,2:10]),
                        as.factor(data_mode_imp[training,11]),
                        type = "C-svc", # Use C-classification method
                        kernel = "vanilladot", # Use simple linear kernel
                        C = amounts[i])

  # Compare models using validation set.

  pred <- predict(model_svm, data_mode_imp[validation,2:10])
  acc_svm[i] = sum(pred == data_mode_imp[validation,11]) / nrow(data_mode_imp[validation,])
}
```

```
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
```

```
# Data with regression imputation

for (i in 1:6) {

  # Fit model using training set.

  model_svm <- ksvm(as.matrix(data_reg_imp[training,2:10]),
                      as.factor(data_reg_imp[training,11]),
                      type = "C-svc", # Use C-classification method
                      kernel = "vanilladot", # Use simple linear kernel
                      C = amounts[i])

  # Compare models using validation set.

  pred <- predict(model_svm, data_reg_imp[validation,2:10])
```

```
  acc_svm[i+6] = sum(pred == data_reg_imp[validation,11]) / nrow(data_reg_imp[validation,])
}
```

```
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
```

```
# Data with regression with perturbation imputation

for (i in 1:6) {

  # Fit model using training set.

  model_svm <- ksvm(as.matrix(data_reg_pert_imp[training,2:10]),
                    as.factor(data_reg_pert_imp[training,11]),
                    type = "C-svc", # Use C-classification method
                    kernel = "vanilladot", # Use simple linear kernel
                    C = amounts[i])

  # Compare models using validation set.

  pred <- predict(model_svm, data_reg_pert_imp[validation,2:10])
  acc_svm[i+12] = sum(pred == data_reg_pert_imp[validation,11]) / nrow(data_reg_pert_imp[validation,])
}
```

```
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
```

```
# Data without missing variables

for (i in 1:6) {

  # Fit model using training set.

  model_svm <- ksvm(as.matrix(data_no_missing[training_no_missing,2:10]),
                    as.factor(data_no_missing[training_no_missing,11]),
                    type = "C-svc", # Use C-classification method
                    kernel = "vanilladot", # Use simple linear kernel
                    C = amounts[i])

  # Compare models using validation set.

  pred <- predict(model_svm, data_no_missing[validation_no_missing,2:10])
  acc_svm[i+18] = sum(pred == data_no_missing[validation_no_missing,11]) /
    nrow(data[validation_no_missing,])
}
```

```
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
```

```
# Data with binary variable to indicate if an observation
# has a missing V7 value. Use the interaction factor for modeling

for (i in 1:6) {

  # Fit model using training set.

  model_svm <- ksvm(as.matrix(data_binary[training,c(2:6,8:10,13)]),
                    as.factor(data_binary[training,11]),
                    type = "C-svc", # Use C-classification method
                    kernel = "vanilladot", # Use simple linear kernel
                    C = amounts[i])

  # Compare models using validation set.

  pred <- predict(model_svm, data_binary[validation,c(2:6,8:10,13)])
  acc_svm[i+24] = sum(pred == data_binary[validation,11]) / nrow(data_binary[validation,])
}
```

```
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
```

```
acc_svm
```

```
##  [1] 0.657 0.948 0.971 0.957 0.957 0.957 0.657 0.948 0.971 0.957 0.957 0.957
## [13] 0.657 0.948 0.971 0.957 0.957 0.957 0.652 0.951 0.975 0.961 0.956 0.956
## [25] 0.657 0.948 0.971 0.957 0.957 0.957
```