VytKra Mathematical Methods for Artificial Intelligence Lab 1

# Reading Data

set.seed(123)  
  
data <- read.csv("superconductor\_dataset.csv")

# Required packages

library(SmartEDA)  
library(dplyr)  
library(purrr)  
library(ggplot2)  
library(corrplot)  
library(caret)  
library(randomForest)

# EDA, first look at the dataset

ExpData(data,type=1)

## Descriptions Value  
## 1 Sample size (nrow) 21263  
## 2 No. of variables (ncol) 169  
## 3 No. of numeric/interger variables 168  
## 4 No. of factor variables 0  
## 5 No. of text variables 1  
## 6 No. of logical variables 0  
## 7 No. of identifier variables 0  
## 8 No. of date variables 0  
## 9 No. of zero variance variables (uniform) 9  
## 10 %. of variables having complete cases 100% (169)  
## 11 %. of variables having >0% and <50% missing cases 0% (0)  
## 12 %. of variables having >=50% and <90% missing cases 0% (0)  
## 13 %. of variables having >=90% missing cases 0% (0)

We have 168 numeric variables and only 1 text variable, no missing values. There are 9 columns with zero variance, we’ll look at it later.

Let’s look at the text variable:

head(data$material)

## [1] "Ba0.2La1.8Cu1O4" "Ba0.1La1.9Ag0.1Cu0.9O4" "Ba0.1La1.9Cu1O4"   
## [4] "Ba0.15La1.85Cu1O4" "Ba0.3La1.7Cu1O4" "Ba0.5La1.5Cu1O4"

Out of total 21263 observations, we have 15542 unique record in this text column. We won’t be using this variable in our analysis.

data <- data %>% select(-material)  
  
# Return a character vector of variable names which have 0 variance  
variables\_with\_zero\_var <- names(data)[which(map\_dbl(data, var) == 0)]  
  
summary(data[,variables\_with\_zero\_var])

## He Ne Ar Kr Xe Pm   
## Min. :0 Min. :0 Min. :0 Min. :0 Min. :0 Min. :0   
## 1st Qu.:0 1st Qu.:0 1st Qu.:0 1st Qu.:0 1st Qu.:0 1st Qu.:0   
## Median :0 Median :0 Median :0 Median :0 Median :0 Median :0   
## Mean :0 Mean :0 Mean :0 Mean :0 Mean :0 Mean :0   
## 3rd Qu.:0 3rd Qu.:0 3rd Qu.:0 3rd Qu.:0 3rd Qu.:0 3rd Qu.:0   
## Max. :0 Max. :0 Max. :0 Max. :0 Max. :0 Max. :0   
## Po At Rn   
## Min. :0 Min. :0 Min. :0   
## 1st Qu.:0 1st Qu.:0 1st Qu.:0   
## Median :0 Median :0 Median :0   
## Mean :0 Mean :0 Mean :0   
## 3rd Qu.:0 3rd Qu.:0 3rd Qu.:0   
## Max. :0 Max. :0 Max. :0

data <- data %>% select(-all\_of(variables\_with\_zero\_var))

Those 9 variables have zero variance, we’ll exclude them

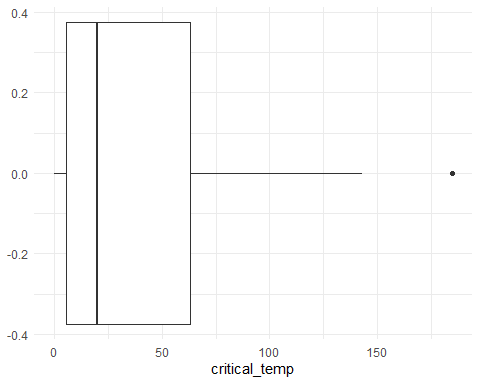
We should look into our target variable - critical\_temp

summary( data %>% select(critical\_temp) )

## critical\_temp   
## Min. : 0.00021   
## 1st Qu.: 5.36500   
## Median : 20.00000   
## Mean : 34.42122   
## 3rd Qu.: 63.00000   
## Max. :185.00000

We see a maximum value of critical\_temp to be 185, while 3rd Quantile - 63. Let’s look at the boxplot:

ggplot(data, aes(x=critical\_temp)) +  
 geom\_boxplot() +  
 theme\_minimal()

 Seems like we have one outlier at >150. We will remove this observation

data <- data %>% filter(critical\_temp < 150)

ExpNumStat(data,by ="A",Outlier=TRUE,round= 2, gp = "critical\_temp") %>%   
 select(Vname, min, max, mean, median, SD, nOutliers)

## Vname min max mean median SD  
## 123 Ag 0.00 7.00 0.01 0.00 0.17  
## 91 Al 0.00 99.92 0.06 0.00 1.13  
## 110 As 0.00 18.00 0.16 0.00 1.08  
## 153 Au 0.00 64.00 0.02 0.00 0.72  
## 84 B 0.00 105.00 0.14 0.00 1.04  
## 131 Ba 0.00 24.00 0.57 0.00 0.98  
## 83 Be 0.00 40.00 0.03 0.00 0.85  
## 157 Bi 0.00 14.00 0.20 0.00 0.66  
## 112 Br 0.00 5.00 0.00 0.00 0.08  
## 85 C 0.00 120.00 0.38 0.00 4.41  
## 97 Ca 0.00 24.00 0.26 0.00 0.90  
## 124 Cd 0.00 100.00 0.01 0.00 0.69  
## 133 Ce 0.00 5.00 0.03 0.00 0.17  
## 95 Cl 0.00 3.00 0.01 0.00 0.12  
## 104 Co 0.00 35.38 0.04 0.00 0.58  
## 101 Cr 0.00 34.90 0.01 0.00 0.25  
## 80 critical\_temp 0.00 143.00 34.41 20.00 34.24  
## 130 Cs 0.00 3.00 0.00 0.00 0.08  
## 106 Cu 0.00 98.00 1.28 0.90 2.08  
## 140 Dy 0.00 5.00 0.01 0.00 0.10  
## 5 entropy\_atomic\_mass 0.00 1.98 1.17 1.20 0.36  
## 25 entropy\_atomic\_radius 0.00 2.14 1.27 1.33 0.38  
## 35 entropy\_Density 0.00 1.95 1.07 1.09 0.34  
## 45 entropy\_ElectronAffinity 0.00 1.77 1.07 1.14 0.34  
## 15 entropy\_fie 0.00 2.16 1.30 1.36 0.38  
## 55 entropy\_FusionHeat 0.00 2.03 1.09 1.11 0.38  
## 65 entropy\_ThermalConductivity 0.00 1.63 0.73 0.74 0.33  
## 75 entropy\_Valence 0.00 2.14 1.30 1.37 0.39  
## 142 Er 0.00 5.00 0.01 0.00 0.13  
## 137 Eu 0.00 6.00 0.02 0.00 0.15  
## 88 F 0.00 4.00 0.01 0.00 0.13  
## 103 Fe 0.00 30.00 0.15 0.00 0.71  
## 108 Ga 0.00 41.00 0.07 0.00 1.12  
## 138 Gd 0.00 4.00 0.02 0.00 0.16  
## 109 Ge 0.00 46.00 0.08 0.00 1.02  
## 3 gmean\_atomic\_mass 5.32 208.98 71.29 66.36 31.03  
## 23 gmean\_atomic\_radius 48.00 298.00 144.45 142.81 22.09  
## 33 gmean\_Density 1.43 22590.00 3460.85 1339.97 3703.27  
## 43 gmean\_ElectronAffinity 1.50 326.10 54.36 51.47 29.00  
## 13 gmean\_fie 375.50 1313.10 737.46 727.96 78.28  
## 53 gmean\_FusionHeat 0.22 105.00 10.14 5.25 10.07  
## 63 gmean\_ThermalConductivity 0.03 317.88 29.84 14.29 34.06  
## 73 gmean\_Valence 1.00 7.00 3.06 2.62 1.05  
## 81 H 0.00 14.00 0.02 0.00 0.27  
## 146 Hf 0.00 25.00 0.01 0.00 0.21  
## 154 Hg 0.00 8.00 0.04 0.00 0.21  
## 141 Ho 0.00 5.00 0.01 0.00 0.10  
## 129 I 0.00 4.00 0.00 0.00 0.09  
## 115 Y 0.00 9.00 0.18 0.00 0.43  
## 144 Yb 0.00 16.00 0.01 0.00 0.21  
## 125 In 0.00 31.50 0.05 0.00 0.52  
## 151 Ir 0.00 45.00 0.06 0.00 0.86  
## 96 K 0.00 3.30 0.02 0.00 0.14  
## 132 La 0.00 98.00 0.26 0.00 2.32  
## 82 Li 0.00 3.00 0.01 0.00 0.13  
## 145 Lu 0.00 7.00 0.03 0.00 0.28  
## 1 mean\_atomic\_mass 6.94 208.98 87.56 84.92 29.67  
## 21 mean\_atomic\_radius 48.00 298.00 157.99 160.25 20.14  
## 31 mean\_Density 1.43 22590.00 6111.71 5329.09 2846.63  
## 41 mean\_ElectronAffinity 1.50 326.10 76.88 73.10 27.70  
## 11 mean\_fie 375.50 1313.10 769.60 764.90 87.45  
## 51 mean\_FusionHeat 0.22 105.00 14.30 9.30 11.30  
## 61 mean\_ThermalConductivity 0.03 332.50 89.71 96.50 38.51  
## 71 mean\_Valence 1.00 7.00 3.20 2.83 1.04  
## 90 Mg 0.00 12.00 0.03 0.00 0.27  
## 102 Mn 0.00 14.00 0.00 0.00 0.13  
## 118 Mo 0.00 99.99 0.15 0.00 2.08  
## 86 N 0.00 12.80 0.01 0.00 0.15  
## 89 Na 0.00 4.00 0.01 0.00 0.10  
## 117 Nb 0.00 99.98 0.44 0.00 4.85  
## 135 Nd 0.00 6.00 0.04 0.00 0.22  
## 105 Ni 0.00 45.00 0.09 0.00 0.98  
## 87 O 0.00 66.00 3.01 1.00 3.81  
## 150 Os 0.00 10.00 0.02 0.00 0.28  
## 93 P 0.00 20.00 0.03 0.00 0.47  
## 156 Pb 0.00 19.00 0.04 0.00 0.27  
## 122 Pd 0.00 51.00 0.09 0.00 1.55  
## 134 Pr 0.00 185.00 0.04 0.00 1.28  
## 152 Pt 0.00 5.80 0.03 0.00 0.31  
## 7 range\_atomic\_mass 0.00 207.97 115.61 122.91 54.63  
## 27 range\_atomic\_radius 0.00 256.00 139.33 171.00 67.27  
## 37 range\_Density 0.00 22588.57 8665.75 8958.57 4096.97  
## 47 range\_ElectronAffinity 0.00 349.00 120.73 127.05 58.70  
## 17 range\_fie 0.00 1304.50 572.23 764.10 309.62  
## 57 range\_FusionHeat 0.00 104.78 21.14 12.88 20.37  
## 67 range\_ThermalConductivity 0.00 429.97 250.91 399.80 158.70  
## 113 Rb 0.00 4.00 0.01 0.00 0.12  
## 149 Re 0.00 97.24 0.04 0.00 1.18  
## 121 Rh 0.00 45.00 0.07 0.00 1.01  
## 120 Ru 0.00 64.00 0.06 0.00 0.77  
## 94 S 0.00 15.00 0.11 0.00 0.76  
## 127 Sb 0.00 83.50 0.10 0.00 1.84  
## 98 Sc 0.00 5.00 0.01 0.00 0.19  
## 111 Se 0.00 19.00 0.08 0.00 0.68  
## 92 Si 0.00 100.00 0.19 0.00 2.22  
## 136 Sm 0.00 12.00 0.02 0.00 0.18  
## 126 Sn 0.00 99.20 0.12 0.00 1.89  
## 114 Sr 0.00 16.70 0.33 0.00 0.76  
## 9 std\_atomic\_mass 0.00 101.02 44.39 45.12 20.03  
## 29 std\_atomic\_radius 0.00 115.50 51.60 58.66 22.90  
## 39 std\_Density 0.00 10724.37 3417.03 3301.89 1673.58  
## 49 std\_ElectronAffinity 0.00 162.90 48.91 51.13 21.74  
## 19 std\_fie 0.00 499.67 215.63 266.37 109.97  
## 59 std\_FusionHeat 0.00 51.63 8.32 4.95 8.67  
## 69 std\_ThermalConductivity 0.00 214.99 98.95 135.76 60.14  
## 78 std\_Valence 0.00 3.00 0.84 0.80 0.48  
## 147 Ta 0.00 55.00 0.04 0.00 0.85  
## 139 Tb 0.00 5.00 0.00 0.00 0.06  
## 119 Tc 0.00 6.00 0.00 0.00 0.06  
## 128 Te 0.00 66.70 0.04 0.00 0.72  
## 99 Ti 0.00 75.00 0.16 0.00 2.73  
## 155 Tl 0.00 7.00 0.05 0.00 0.27  
## 143 Tm 0.00 5.00 0.01 0.00 0.13  
## 100 V 0.00 79.50 0.22 0.00 3.41  
## 148 W 0.00 14.00 0.01 0.00 0.16  
## 6 wtd\_entropy\_atomic\_mass 0.00 1.96 1.06 1.15 0.40  
## 26 wtd\_entropy\_atomic\_radius 0.00 1.90 1.13 1.24 0.41  
## 36 wtd\_entropy\_Density 0.00 1.70 0.86 0.88 0.32  
## 46 wtd\_entropy\_ElectronAffinity 0.00 1.68 0.77 0.78 0.29  
## 16 wtd\_entropy\_fie 0.00 2.04 0.93 0.92 0.33  
## 56 wtd\_entropy\_FusionHeat 0.00 1.75 0.91 0.99 0.37  
## 66 wtd\_entropy\_ThermalConductivity 0.00 1.61 0.54 0.55 0.32  
## 76 wtd\_entropy\_Valence 0.00 1.95 1.05 1.17 0.38  
## 4 wtd\_gmean\_atomic\_mass 1.96 208.98 58.54 39.92 36.65  
## 24 wtd\_gmean\_atomic\_radius 48.00 298.00 120.99 113.18 35.84  
## 34 wtd\_gmean\_Density 0.69 22590.00 3117.39 1515.53 3975.16  
## 44 wtd\_gmean\_ElectronAffinity 1.50 326.10 72.41 73.17 31.65  
## 14 wtd\_gmean\_fie 375.50 1327.59 832.75 856.19 119.75  
## 54 wtd\_gmean\_FusionHeat 0.22 105.00 10.14 4.93 13.13  
## 64 wtd\_gmean\_ThermalConductivity 0.02 376.03 27.31 6.10 40.19  
## 74 wtd\_gmean\_Valence 1.00 7.00 3.06 2.43 1.17  
## 2 wtd\_mean\_atomic\_mass 6.42 208.98 72.99 60.70 33.49  
## 22 wtd\_mean\_atomic\_radius 48.00 298.00 134.72 125.97 28.80  
## 32 wtd\_mean\_Density 1.43 22590.00 5267.41 4303.71 3221.23  
## 42 wtd\_mean\_ElectronAffinity 1.50 326.10 92.72 102.86 32.28  
## 12 wtd\_mean\_fie 375.50 1348.03 870.43 889.96 143.26  
## 52 wtd\_mean\_FusionHeat 0.22 105.00 13.85 8.33 14.28  
## 62 wtd\_mean\_ThermalConductivity 0.03 406.96 81.55 73.33 45.52  
## 72 wtd\_mean\_Valence 1.00 7.00 3.15 2.62 1.19  
## 8 wtd\_range\_atomic\_mass 0.00 205.59 33.23 26.64 26.97  
## 28 wtd\_range\_atomic\_radius 0.00 240.16 51.37 43.00 35.02  
## 38 wtd\_range\_Density 0.00 22434.16 2902.84 2082.96 2398.48  
## 48 wtd\_range\_ElectronAffinity 0.00 218.70 59.33 71.16 28.62  
## 18 wtd\_range\_fie 0.00 1251.86 483.51 510.44 224.05  
## 58 wtd\_range\_FusionHeat 0.00 102.67 8.22 3.44 11.41  
## 68 wtd\_range\_ThermalConductivity 0.00 401.44 62.04 56.56 43.12  
## 77 wtd\_range\_Valence 0.00 6.99 1.48 1.06 0.98  
## 10 wtd\_std\_atomic\_mass 0.00 101.02 41.45 44.29 19.98  
## 30 wtd\_std\_atomic\_radius 0.00 97.14 52.34 59.94 25.29  
## 40 wtd\_std\_Density 0.00 10410.93 3319.28 3625.63 1611.75  
## 50 wtd\_std\_ElectronAffinity 0.00 169.08 44.41 48.03 20.43  
## 20 wtd\_std\_fie 0.00 479.16 224.05 258.46 127.93  
## 60 wtd\_std\_FusionHeat 0.00 51.68 7.72 5.50 7.29  
## 70 wtd\_std\_ThermalConductivity 0.00 213.30 96.24 113.56 63.71  
## 79 wtd\_std\_Valence 0.00 3.00 0.67 0.50 0.46  
## 107 Zn 0.00 20.00 0.01 0.00 0.40  
## 116 Zr 0.00 96.71 0.37 0.00 4.85  
## nOutliers  
## 123 156  
## 91 731  
## 110 1502  
## 153 242  
## 84 1205  
## 131 167  
## 83 96  
## 157 2389  
## 112 66  
## 85 1274  
## 97 4112  
## 124 178  
## 133 1162  
## 95 146  
## 104 1035  
## 101 195  
## 80 0  
## 130 98  
## 106 72  
## 140 239  
## 5 386  
## 25 285  
## 35 601  
## 45 567  
## 15 285  
## 55 285  
## 65 0  
## 75 285  
## 142 356  
## 137 380  
## 88 669  
## 103 2339  
## 108 605  
## 138 663  
## 109 520  
## 3 3313  
## 23 1502  
## 33 494  
## 43 821  
## 13 1479  
## 53 1485  
## 63 1245  
## 73 494  
## 81 298  
## 146 222  
## 154 845  
## 141 227  
## 129 83  
## 115 4075  
## 144 291  
## 125 544  
## 151 572  
## 96 530  
## 132 3463  
## 82 311  
## 145 399  
## 1 1605  
## 21 1202  
## 31 1827  
## 41 2229  
## 11 1912  
## 51 1562  
## 61 300  
## 71 7  
## 90 522  
## 102 171  
## 118 888  
## 86 306  
## 89 322  
## 117 1436  
## 135 946  
## 105 1149  
## 87 50  
## 150 255  
## 93 355  
## 156 1255  
## 122 487  
## 134 1276  
## 152 419  
## 7 0  
## 27 0  
## 37 2908  
## 47 1524  
## 17 0  
## 57 2112  
## 67 0  
## 113 158  
## 149 360  
## 121 643  
## 120 735  
## 94 693  
## 127 343  
## 98 149  
## 111 685  
## 92 725  
## 136 478  
## 126 840  
## 114 4852  
## 9 4  
## 29 0  
## 39 3496  
## 49 2518  
## 19 0  
## 59 2090  
## 69 0  
## 78 43  
## 147 396  
## 139 104  
## 119 50  
## 128 527  
## 99 589  
## 155 908  
## 143 181  
## 100 798  
## 148 265  
## 6 0  
## 26 0  
## 36 737  
## 46 3428  
## 16 2176  
## 56 0  
## 66 3  
## 76 0  
## 4 1247  
## 24 8  
## 34 439  
## 44 348  
## 14 4  
## 54 577  
## 64 863  
## 74 52  
## 2 1239  
## 22 11  
## 32 926  
## 42 528  
## 12 0  
## 52 1173  
## 62 1420  
## 72 30  
## 8 1629  
## 28 1879  
## 38 1657  
## 48 170  
## 18 0  
## 58 1374  
## 68 196  
## 77 1394  
## 10 15  
## 30 0  
## 40 2261  
## 50 1458  
## 20 0  
## 60 2643  
## 70 0  
## 79 24  
## 107 832  
## 116 631

We should look at the correlation between variables

# Correlation  
  
corr\_simple <- function(df,sig=0.5){  
 corr <- cor(df)  
 #prepare to drop duplicates and correlations of 1   
 corr[lower.tri(corr,diag=TRUE)] <- NA   
 #drop perfect correlations  
 corr[corr == 1] <- NA   
 #turn into a 3-column table  
 corr <- as.data.frame(as.table(corr))  
 #remove the NA values from above   
 corr <- na.omit(corr)   
 #select significant values   
 corr <- subset(corr, abs(Freq) > sig)   
 #sort by highest correlation  
 corr <- corr[order(-abs(corr$Freq)),]   
 return(corr)  
}  
  
correlation\_matrix = cor(data)

We have 4 variables with higher than .99 correlation. We have 22 variables with higher than .95 correlation. We have 35 variables with higher than .9 correlation.

# Pre-process

## Data split

indices <- createDataPartition(data$critical\_temp, p = 0.8, list = FALSE)  
train <- data[indices,]  
test <- data[-indices,]

Splitted a dataset by 80% / 20% rule. Created a training dataset with 17010 (80.0018813%) observations and testing dataset with 4252 (19.9981187%) observations.

## Data transformation

preProcValues <- preProcess(train, method = c("center", "scale"))  
train\_transformed <- predict(preProcValues, train)  
test\_transformed <- predict(preProcValues, test)  
  
label\_index <- which(colnames(train\_transformed) == "critical\_temp")

We also centered and scaled our data. We transformed the testing dataset based on the pre process of training dataset.

# ML algorithms

## Linear Regression

# linear regression  
train\_control\_cv <- trainControl(method = "cv", number = 10)  
fit\_lm <- train(critical\_temp ~ ., data = train\_transformed, method = "lm", trControl = train\_control\_cv)

We are performing linear regression model on the whole training dataset with 10 fold cross-validation. The model:

print(fit\_lm)

## Linear Regression   
##   
## 17010 samples  
## 158 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 15309, 15309, 15310, 15309, 15309, 15307, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 0.5476384 0.7229608 0.3632821  
##   
## Tuning parameter 'intercept' was held constant at a value of TRUE

Using fitted linear regression model on the testing dataset:

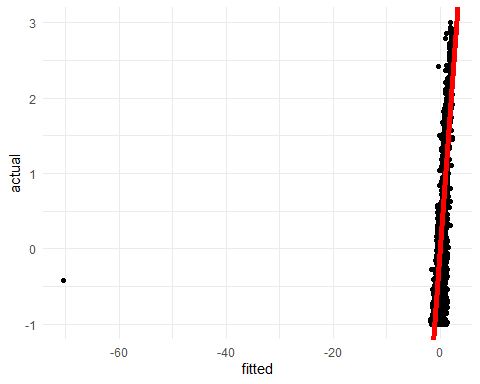
test\_lm <- predict(fit\_lm, newdata = test\_transformed)  
print(round(postResample(pred = test\_lm, obs = test\_transformed$critical\_temp), 3))

## RMSE Rsquared MAE   
## 1.180 0.306 0.380

Our model, on the testing dataset, only explains ~30% of variance, while on a training set the rsquared was 72%. But the MAE was 0.36 for the training set and 0.38 for the testing dataset, it could mean an outlier influences the RMSE and rqsuared.

Let’s look at the fitted vs actual plot:

fitted\_actual\_lm <- data.frame(fitted = test\_lm, actual = test\_transformed$critical\_temp)  
  
ggplot(fitted\_actual\_lm,   
 aes(x = fitted,  
 y = actual)) +  
 geom\_point() +  
 geom\_abline(intercept = 0,  
 slope = 1,  
 color = "red",  
 size = 2) +  
 theme\_minimal()

 We clearly see one value, with an actual value of ~-0.5, while the fitted value is ~-70.

Let’s look at that observation before transformation:

test[test\_lm < -60,] %>%   
 tidyr::pivot\_longer(cols = everything())

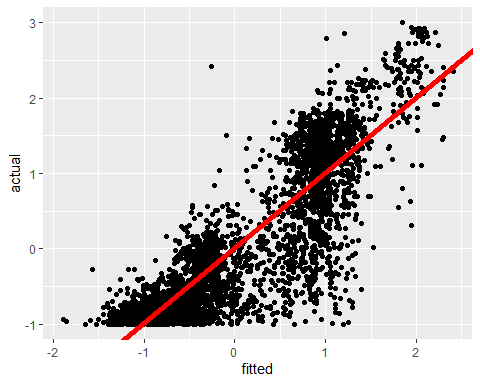
## # A tibble: 159 x 2  
## name value  
## <chr> <dbl>  
## 1 number\_of\_elements 4   
## 2 mean\_atomic\_mass 90.1   
## 3 wtd\_mean\_atomic\_mass 138.   
## 4 gmean\_atomic\_mass 66.9   
## 5 wtd\_gmean\_atomic\_mass 134.   
## 6 entropy\_atomic\_mass 1.18   
## 7 wtd\_entropy\_atomic\_mass 0.0406  
## 8 range\_atomic\_mass 125.   
## 9 wtd\_range\_atomic\_mass 137.   
## 10 std\_atomic\_mass 53.1   
## # ... with 149 more rows

If we would exclude this observation from our testing dataset:

# without the outlier  
print(round(postResample(pred = test\_lm[test\_lm > -60], obs = test\_transformed$critical\_temp[test\_lm > -60]), 3))

## RMSE Rsquared MAE   
## 0.489 0.759 0.364

ggplot(fitted\_actual\_lm %>% filter(fitted > -60),   
 aes(x = fitted,  
 y = actual)) +  
 geom\_point() +  
 geom\_abline(intercept = 0,  
 slope = 1,  
 color = "red",  
 size = 2)

 Our rsquared would be 76%.

## Linear Regression with removed correlated variables

# Removed >.9 correlations  
correlation\_matrix = cor(train\_transformed)  
correlated\_columns = findCorrelation(correlation\_matrix, cutoff = 0.9)  
correlated\_columns = sort(correlated\_columns)  
  
colnames(train\_transformed[, correlated\_columns])

## [1] "number\_of\_elements" "gmean\_atomic\_mass"   
## [3] "wtd\_gmean\_atomic\_mass" "entropy\_atomic\_mass"   
## [5] "range\_atomic\_mass" "wtd\_std\_atomic\_mass"   
## [7] "mean\_fie" "wtd\_mean\_fie"   
## [9] "entropy\_fie" "range\_fie"   
## [11] "wtd\_std\_fie" "gmean\_atomic\_radius"   
## [13] "wtd\_gmean\_atomic\_radius" "entropy\_atomic\_radius"   
## [15] "wtd\_entropy\_atomic\_radius" "range\_atomic\_radius"   
## [17] "wtd\_std\_atomic\_radius" "wtd\_gmean\_Density"   
## [19] "range\_Density" "wtd\_std\_Density"   
## [21] "range\_ElectronAffinity" "wtd\_mean\_FusionHeat"   
## [23] "gmean\_FusionHeat" "wtd\_gmean\_FusionHeat"   
## [25] "entropy\_FusionHeat" "std\_FusionHeat"   
## [27] "wtd\_std\_FusionHeat" "range\_ThermalConductivity"   
## [29] "wtd\_std\_ThermalConductivity" "mean\_Valence"   
## [31] "wtd\_mean\_Valence" "wtd\_gmean\_Valence"   
## [33] "entropy\_Valence" "wtd\_entropy\_Valence"   
## [35] "range\_Valence"

We are going to remove these columns from our training dataset, those columns have >.9 correlation

train\_transformed\_no\_correlated = train\_transformed[,-c(correlated\_columns)]  
label\_index\_noCorr <- which(colnames(train\_transformed\_no\_correlated) == "critical\_temp")  
fit\_lm\_no\_correlated <- train(critical\_temp ~ ., data = train\_transformed\_no\_correlated, method = "lm", trControl = train\_control\_cv)

Performing Linear Regression with 10 fold cross-validation

We see, that our rsquared on training dataset dropped to 69%.

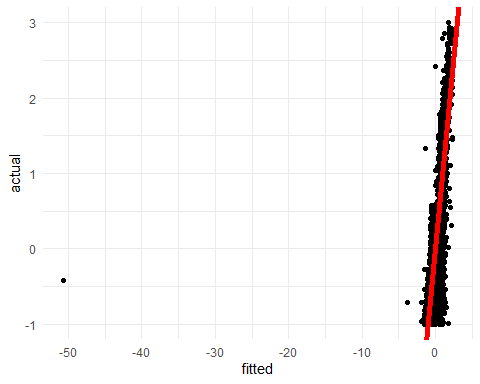
Testing on a test dataset

test\_lm\_noCorr <- predict(fit\_lm\_no\_correlated, newdata = test\_transformed[, -c(correlated\_columns, label\_index)])  
round(postResample(pred = test\_lm\_noCorr, obs = test\_transformed$critical\_temp), 3)

## RMSE Rsquared MAE   
## 0.925 0.416 0.391

Rsquared on the testing dataset is 41%, we see similar results with MAE, let’s look at the plot

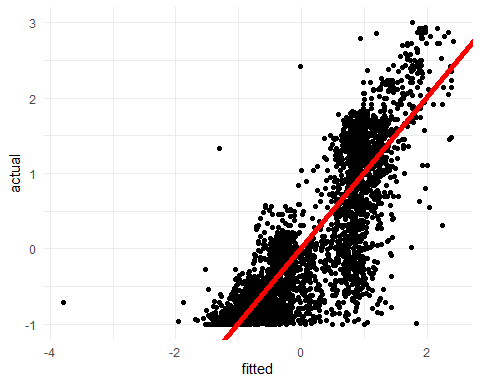
fitted\_actual\_lm\_noCorr <- data.frame(fitted = test\_lm\_noCorr, actual = test\_transformed$critical\_temp)  
  
ggplot(fitted\_actual\_lm\_noCorr,   
 aes(x = fitted,  
 y = actual)) +  
 geom\_point() +  
 geom\_abline(intercept = 0,  
 slope = 1,  
 color = "red",  
 size = 2) +  
 theme\_minimal()

 Same result with that one outlier, let’s try without it:

round(postResample(pred = test\_lm\_noCorr[test\_lm\_noCorr > -50], obs = test\_transformed$critical\_temp[test\_lm\_noCorr > -50]), 3)

## RMSE Rsquared MAE   
## 0.509 0.739 0.379

ggplot(fitted\_actual\_lm\_noCorr %>% filter(fitted > -50),   
 aes(x = fitted,  
 y = actual)) +  
 geom\_point() +  
 geom\_abline(intercept = 0,  
 slope = 1,  
 color = "red",  
 size = 2) +  
 theme\_minimal()

 The rsquared increased to 74%. Slightly worse than using all variables.

# Random Forest

Out of curiosity, let’s try a simple random forest model to see how it fits our data.

fit\_rf <- randomForest(x = train\_transformed[, -label\_index], y = train\_transformed[, label\_index], ntree = 150, do.trace = T)

## | Out-of-bag |  
## Tree | MSE %Var(y) |  
## 1 | 0.1387 13.87 |  
## 2 | 0.1304 13.04 |  
## 3 | 0.1219 12.19 |  
## 4 | 0.1173 11.73 |  
## 5 | 0.1154 11.54 |  
## 6 | 0.1103 11.03 |  
## 7 | 0.1052 10.52 |  
## 8 | 0.09959 9.96 |  
## 9 | 0.09633 9.63 |  
## 10 | 0.09343 9.34 |  
## 11 | 0.09052 9.05 |  
## 12 | 0.08938 8.94 |  
## 13 | 0.0873 8.73 |  
## 14 | 0.08605 8.61 |  
## 15 | 0.08486 8.49 |  
## 16 | 0.08453 8.45 |  
## 17 | 0.08366 8.37 |  
## 18 | 0.08285 8.29 |  
## 19 | 0.08215 8.22 |  
## 20 | 0.08174 8.17 |  
## 21 | 0.08114 8.11 |  
## 22 | 0.08093 8.09 |  
## 23 | 0.08044 8.04 |  
## 24 | 0.08001 8.00 |  
## 25 | 0.0794 7.94 |  
## 26 | 0.07899 7.90 |  
## 27 | 0.07862 7.86 |  
## 28 | 0.07838 7.84 |  
## 29 | 0.07809 7.81 |  
## 30 | 0.07779 7.78 |  
## 31 | 0.07724 7.72 |  
## 32 | 0.07686 7.69 |  
## 33 | 0.0768 7.68 |  
## 34 | 0.0765 7.65 |  
## 35 | 0.07621 7.62 |  
## 36 | 0.07602 7.60 |  
## 37 | 0.07575 7.58 |  
## 38 | 0.07561 7.56 |  
## 39 | 0.07547 7.55 |  
## 40 | 0.07537 7.54 |  
## 41 | 0.07539 7.54 |  
## 42 | 0.07529 7.53 |  
## 43 | 0.07529 7.53 |  
## 44 | 0.07524 7.52 |  
## 45 | 0.07519 7.52 |  
## 46 | 0.07496 7.50 |  
## 47 | 0.07479 7.48 |  
## 48 | 0.07464 7.46 |  
## 49 | 0.07449 7.45 |  
## 50 | 0.07452 7.45 |  
## 51 | 0.07458 7.46 |  
## 52 | 0.07448 7.45 |  
## 53 | 0.07434 7.43 |  
## 54 | 0.07429 7.43 |  
## 55 | 0.07422 7.42 |  
## 56 | 0.07422 7.42 |  
## 57 | 0.07414 7.41 |  
## 58 | 0.0741 7.41 |  
## 59 | 0.07402 7.40 |  
## 60 | 0.07401 7.40 |  
## 61 | 0.074 7.40 |  
## 62 | 0.074 7.40 |  
## 63 | 0.07401 7.40 |  
## 64 | 0.07383 7.38 |  
## 65 | 0.07377 7.38 |  
## 66 | 0.07366 7.37 |  
## 67 | 0.07361 7.36 |  
## 68 | 0.07357 7.36 |  
## 69 | 0.07351 7.35 |  
## 70 | 0.07352 7.35 |  
## 71 | 0.07348 7.35 |  
## 72 | 0.07333 7.33 |  
## 73 | 0.07336 7.34 |  
## 74 | 0.07334 7.33 |  
## 75 | 0.07327 7.33 |  
## 76 | 0.07324 7.32 |  
## 77 | 0.07317 7.32 |  
## 78 | 0.07308 7.31 |  
## 79 | 0.07295 7.30 |  
## 80 | 0.07287 7.29 |  
## 81 | 0.07277 7.28 |  
## 82 | 0.07265 7.27 |  
## 83 | 0.0726 7.26 |  
## 84 | 0.0726 7.26 |  
## 85 | 0.07257 7.26 |  
## 86 | 0.07258 7.26 |  
## 87 | 0.07255 7.26 |  
## 88 | 0.07256 7.26 |  
## 89 | 0.0725 7.25 |  
## 90 | 0.07245 7.25 |  
## 91 | 0.07238 7.24 |  
## 92 | 0.07236 7.24 |  
## 93 | 0.07234 7.23 |  
## 94 | 0.0723 7.23 |  
## 95 | 0.07223 7.22 |  
## 96 | 0.07222 7.22 |  
## 97 | 0.07222 7.22 |  
## 98 | 0.07217 7.22 |  
## 99 | 0.07219 7.22 |  
## 100 | 0.07214 7.21 |  
## 101 | 0.07215 7.22 |  
## 102 | 0.07213 7.21 |  
## 103 | 0.07215 7.21 |  
## 104 | 0.07211 7.21 |  
## 105 | 0.07209 7.21 |  
## 106 | 0.07208 7.21 |  
## 107 | 0.07204 7.20 |  
## 108 | 0.07202 7.20 |  
## 109 | 0.07203 7.20 |  
## 110 | 0.07201 7.20 |  
## 111 | 0.07194 7.19 |  
## 112 | 0.07191 7.19 |  
## 113 | 0.07192 7.19 |  
## 114 | 0.0719 7.19 |  
## 115 | 0.07186 7.19 |  
## 116 | 0.07184 7.18 |  
## 117 | 0.07183 7.18 |  
## 118 | 0.0718 7.18 |  
## 119 | 0.07174 7.17 |  
## 120 | 0.07174 7.17 |  
## 121 | 0.07169 7.17 |  
## 122 | 0.07168 7.17 |  
## 123 | 0.07162 7.16 |  
## 124 | 0.07162 7.16 |  
## 125 | 0.07161 7.16 |  
## 126 | 0.07161 7.16 |  
## 127 | 0.07162 7.16 |  
## 128 | 0.07163 7.16 |  
## 129 | 0.07159 7.16 |  
## 130 | 0.07156 7.16 |  
## 131 | 0.07152 7.15 |  
## 132 | 0.07149 7.15 |  
## 133 | 0.07147 7.15 |  
## 134 | 0.07139 7.14 |  
## 135 | 0.07143 7.14 |  
## 136 | 0.07144 7.14 |  
## 137 | 0.07142 7.14 |  
## 138 | 0.07141 7.14 |  
## 139 | 0.07136 7.14 |  
## 140 | 0.07135 7.14 |  
## 141 | 0.07133 7.13 |  
## 142 | 0.07135 7.14 |  
## 143 | 0.07135 7.14 |  
## 144 | 0.07132 7.13 |  
## 145 | 0.07136 7.14 |  
## 146 | 0.07138 7.14 |  
## 147 | 0.07137 7.14 |  
## 148 | 0.07134 7.13 |  
## 149 | 0.07134 7.13 |  
## 150 | 0.07131 7.13 |

print(fit\_rf)

##   
## Call:  
## randomForest(x = train\_transformed[, -label\_index], y = train\_transformed[, label\_index], ntree = 150, do.trace = T)   
## Type of random forest: regression  
## Number of trees: 150  
## No. of variables tried at each split: 52  
##   
## Mean of squared residuals: 0.07130602  
## % Var explained: 92.87

Our random forest model on training dataset explains almost 93% or variance, let’s try on a test dataset

test\_rf = predict(fit\_rf, newdata = test\_transformed[, -label\_index])  
round(postResample(pred = test\_rf, obs = test\_transformed$critical\_temp), 3)

## RMSE Rsquared MAE   
## 0.258 0.933 0.150

Rsquared on a test dataset is 93%, good results. :)

fitted\_actual\_rf <- data.frame(fitted = test\_rf, actual = test\_transformed$critical\_temp)  
  
ggplot(fitted\_actual\_rf,   
 aes(x = fitted,  
 y = actual)) +  
 geom\_point() +  
 geom\_abline(intercept = 0,  
 slope = 1,  
 color = "red",  
 size = 2) +  
 theme\_minimal()

