Superconductor Analysis

Predicting the Critical Temperature of a Superconductor

- Reyash Kadyan

Programming Language: R 3.5.1 in Jupyter Notebook

R Libraries used:

- dplyr
- reshape2
- ggplot2
- glmnet
- xgboost
- GGally
- praznik
- caret
- car

Table of Contents

1. Introduction

Superconductivity is a phenomenon of exactly zero electrical resistance and expulsion of magnetic flux fields occurring in certain materials, called superconductors, when cooled below a characteristic critical temperature. Superconductors are widely used in many industry fields, e.g. the Magnetic Resonance Imaging (MRI) in health care, electricity transportation in energy industry and magnetic separation, etc.

Predicting the critical temperature (Tc) of a superconductor is still an open problem in the scientific community. In the past, simple empirical rules based on experiments have guided researchers in synthesizing superconducting materials for many years. Nowadays, features (or predictors) based on the superconductor's elemental properties can be generated and used to predict Tc. In this project, we are going to analyze superconductor data from the Superconducting Material Database maintained by Japan's National Institute for Materials Science (NIMS). The aim is to build statistical models that can predict Tc based on the material's chemical properties.

Specifically, you are going to analyse a superconductor data set, which is based on real world material science data.

Importing libraries

```
In [1]: library(dplyr)
        library(reshape2)
        library(ggplot2)
        library(glmnet)
        library(xgboost)
        library(GGally)
        library(praznik)
        library(caret)
        library(car)
```

```
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

Loading required package: Matrix
Loading required package: foreach
Loaded glmnet 2.0-16


Attaching package: ‘xgboost’

The following object is masked from ‘package:dplyr’:

    slice


Attaching package: ‘GGally’

The following object is masked from ‘package:dplyr’:

    nasa

Loading required package: lattice
Loading required package: carData

Attaching package: ‘car’

The following object is masked from ‘package:dplyr’:

    recode
```

Reading Data

```
In [2]: conduct <- read.csv('train.csv')
```

Let's have a look at the data.

In [3]:
```
head(conduct)
```

number_of_elements	mean_atomic_mass	wtd_mean_atomic_mass	gmean_atomic_mass	wtd_gmean_atomic_mass	entropy_atomic_mass	wtd_entropy_atomic_mass	range_atomic_mass	wtd_range_atomic_mass	std_atomic_m
4	88.94447	57.86269	66.36159	36.11661	1.181795	1.0623955	122.9061	31.79492	51.96
5	92.72921	58.51842	73.13279	36.39660	1.449309	1.0577551	122.9061	36.16194	47.09
4	88.94447	57.88524	66.36159	36.12251	1.181795	0.9759805	122.9061	35.74110	51.96
4	88.94447	57.87397	66.36159	36.11956	1.181795	1.0222909	122.9061	33.76801	51.96
4	88.94447	57.84014	66.36159	36.11072	1.181795	1.1292237	122.9061	27.84874	51.96
4	88.94447	57.79504	66.36159	36.09893	1.181795	1.2252028	122.9061	20.68746	51.96

In [4]:
```
print(paste('Number of rows in data:',dim(conduct)[1]))
print(paste('Number of columns in data:',dim(conduct)[2]))
```

```
[1] "Number of rows in data: 21263"
[1] "Number of columns in data: 82"
```

In [5]:
```
print(paste('Structure of data is:\n\n'))
str(conduct)
```

```
[1] "Structure of data is:\n\n"
'data.frame':   21263 obs. of  82 variables:
 $ number_of_elements             : int  4 5 4 4 4 4 4 4 4 4 ...
 $ mean_atomic_mass               : num  88.9 92.7 88.9 88.9 88.9 ...
 $ wtd_mean_atomic_mass           : num  57.9 58.5 57.9 57.9 57.8 ...
 $ gmean_atomic_mass              : num  66.4 73.1 66.4 66.4 66.4 ...
 $ wtd_gmean_atomic_mass          : num  36.1 36.4 36.1 36.1 36.1 ...
 $ entropy_atomic_mass            : num  1.18 1.45 1.18 1.18 1.18 ...
 $ wtd_entropy_atomic_mass        : num  1.062 1.058 0.976 1.022 1.129 ...
 $ range_atomic_mass              : num  123 123 123 123 123 ...
 $ wtd_range_atomic_mass          : num  31.8 36.2 35.7 33.8 27.8 ...
 $ std_atomic_mass                : num  52 47.1 52 52 52 ...
 $ wtd_std_atomic_mass            : num  53.6 54 53.7 53.6 53.6 ...
 $ mean_fie                       : num  775 766 775 775 775 ...
 $ wtd_mean_fie                   : num  1010 1011 1011 1011 1010 ...
 $ gmean_fie                      : num  718 721 718 718 718 ...
 $ wtd_gmean_fie                  : num  938 939 939 939 937 ...
 $ entropy_fie                    : num  1.31 1.54 1.31 1.31 1.31 ...
 $ wtd_entropy_fie                : num  0.791 0.807 0.774 0.783 0.805 ...
 $ range_fie                      : num  811 811 811 811 811 ...
 $ wtd_range_fie                  : num  736 743 743 740 729 ...
 $ std_fie                        : num  324 290 324 324 324 ...
 $ wtd_std_fie                    : num  356 355 355 355 356 ...
 $ mean_atomic_radius             : num  160 161 160 160 160 ...
 $ wtd_mean_atomic_radius         : num  106 105 105 105 106 ...
 $ gmean_atomic_radius            : num  136 141 136 136 136 ...
 $ wtd_gmean_atomic_radius        : num  84.5 84.4 84.2 84.4 84.8 ...
 $ entropy_atomic_radius          : num  1.26 1.51 1.26 1.26 1.26 ...
 $ wtd_entropy_atomic_radius      : num  1.21 1.2 1.13 1.17 1.26 ...
 $ range_atomic_radius            : int  205 205 205 205 205 205 205 171 171 171 ...
 $ wtd_range_atomic_radius        : num  42.9 50.6 49.3 46.1 36.5 ...
 $ std_atomic_radius              : num  75.2 67.3 75.2 75.2 75.2 ...
 $ wtd_std_atomic_radius          : num  69.2 68 67.8 68.5 70.6 ...
 $ mean_Density                   : num  4654 5821 4654 4654 4654 ...
 $ wtd_mean_Density               : num  2962 3021 2999 2980 2924 ...
 $ gmean_Density                  : num  725 1237 725 725 725 ...
 $ wtd_gmean_Density              : num  53.5 54.1 54 53.8 53.1 ...
 $ entropy_Density                : num  1.03 1.31 1.03 1.03 1.03 ...
 $ wtd_entropy_Density            : num  0.815 0.915 0.76 0.789 0.86 ...
 $ range_Density                  : num  8959 10489 8959 8959 8959 ...
 $ wtd_range_Density              : num  1580 1667 1667 1623 1492 ...
 $ std_Density                    : num  3306 3767 3306 3306 3306 ...
 $ wtd_std_Density                : num  3573 3633 3592 3582 3553 ...
 $ mean_ElectronAffinity          : num  81.8 90.9 81.8 81.8 81.8 ...
 $ wtd_mean_ElectronAffinity      : num  112 112 112 112 111 ...
 $ gmean_ElectronAffinity         : num  60.1 69.8 60.1 60.1 60.1 ...
 $ wtd_gmean_ElectronAffinity     : num  99.4 101.2 101.1 100.2 97.8 ...
 $ entropy_ElectronAffinity       : num  1.16 1.43 1.16 1.16 1.16 ...
 $ wtd_entropy_ElectronAffinity   : num  0.787 0.839 0.786 0.787 0.787 ...
 $ range_ElectronAffinity         : num  127 127 127 127 127 ...
 $ wtd_range_ElectronAffinity     : num  81 81.2 81.2 81.1 80.8 ...
 $ std_ElectronAffinity           : num  51.4 49.4 51.4 51.4 51.4 ...
 $ wtd_std_ElectronAffinity       : num  42.6 41.7 41.6 42.1 43.5 ...
 $ mean_FusionHeat                : num  6.91 7.78 6.91 6.91 6.91 ...
 $ wtd_mean_FusionHeat            : num  3.85 3.8 3.82 3.83 3.87 ...
 $ gmean_FusionHeat               : num  3.48 4.4 3.48 3.48 3.48 ...
 $ wtd_gmean_FusionHeat           : num  1.04 1.04 1.04 1.04 1.04 ...
 $ entropy_FusionHeat             : num  1.09 1.37 1.09 1.09 1.09 ...
 $ wtd_entropy_FusionHeat         : num  0.995 1.073 0.927 0.964 1.045 ...
 $ range_FusionHeat               : num  12.9 12.9 12.9 12.9 12.9 ...
 $ wtd_range_FusionHeat           : num  1.74 1.6 1.76 1.74 1.74 ...
 $ std_FusionHeat                 : num  4.6 4.47 4.6 4.6 4.6 ...
 $ wtd_std_FusionHeat             : num  4.67 4.6 4.65 4.66 4.68 ...
 $ mean_ThermalConductivity       : num  108 172 108 108 108 ...
 $ wtd_mean_ThermalConductivity   : num  61 61.4 60.9 61 61.1 ...
 $ gmean_ThermalConductivity      : num  7.06 16.06 7.06 7.06 7.06 ...
 $ wtd_gmean_ThermalConductivity  : num  0.622 0.62 0.619 0.621 0.625 ...
 $ entropy_ThermalConductivity    : num  0.308 0.847 0.308 0.308 0.308 ...
 $ wtd_entropy_ThermalConductivity: num  0.263 0.568 0.25 0.257 0.273 ...
 $ range_ThermalConductivity      : num  400 430 400 400 400 ...
 $ wtd_range_ThermalConductivity  : num  57.1 51.4 57.1 57.1 57.1 ...
 $ std_ThermalConductivity        : num  169 199 169 169 169 ...
 $ wtd_std_ThermalConductivity    : num  139 140 139 139 138 ...
 $ mean_Valence                   : num  2.25 2 2.25 2.25 2.25 2.25 2.25 2.25 2.25 2.25 ...
 $ wtd_mean_Valence               : num  2.26 2.26 2.27 2.26 2.24 ...
 $ gmean_Valence                  : num  2.21 1.89 2.21 2.21 2.21 ...
 $ wtd_gmean_Valence              : num  2.22 2.21 2.23 2.23 2.21 ...
 $ entropy_Valence                : num  1.37 1.56 1.37 1.37 1.37 ...
 $ wtd_entropy_Valence            : num  1.07 1.05 1.03 1.05 1.1 ...
 $ range_Valence                  : int  1 2 1 1 1 1 1 1 1 1 ...
 $ wtd_range_Valence              : num  1.09 1.13 1.11 1.1 1.06 ...
 $ std_Valence                    : num  0.433 0.632 0.433 0.433 0.433 ...
 $ wtd_std_Valence                : num  0.437 0.469 0.445 0.441 0.429 ...
 $ critical_temp                  : num  29 26 19 22 23 23 11 33 36 31 ...
```

Splitting the data

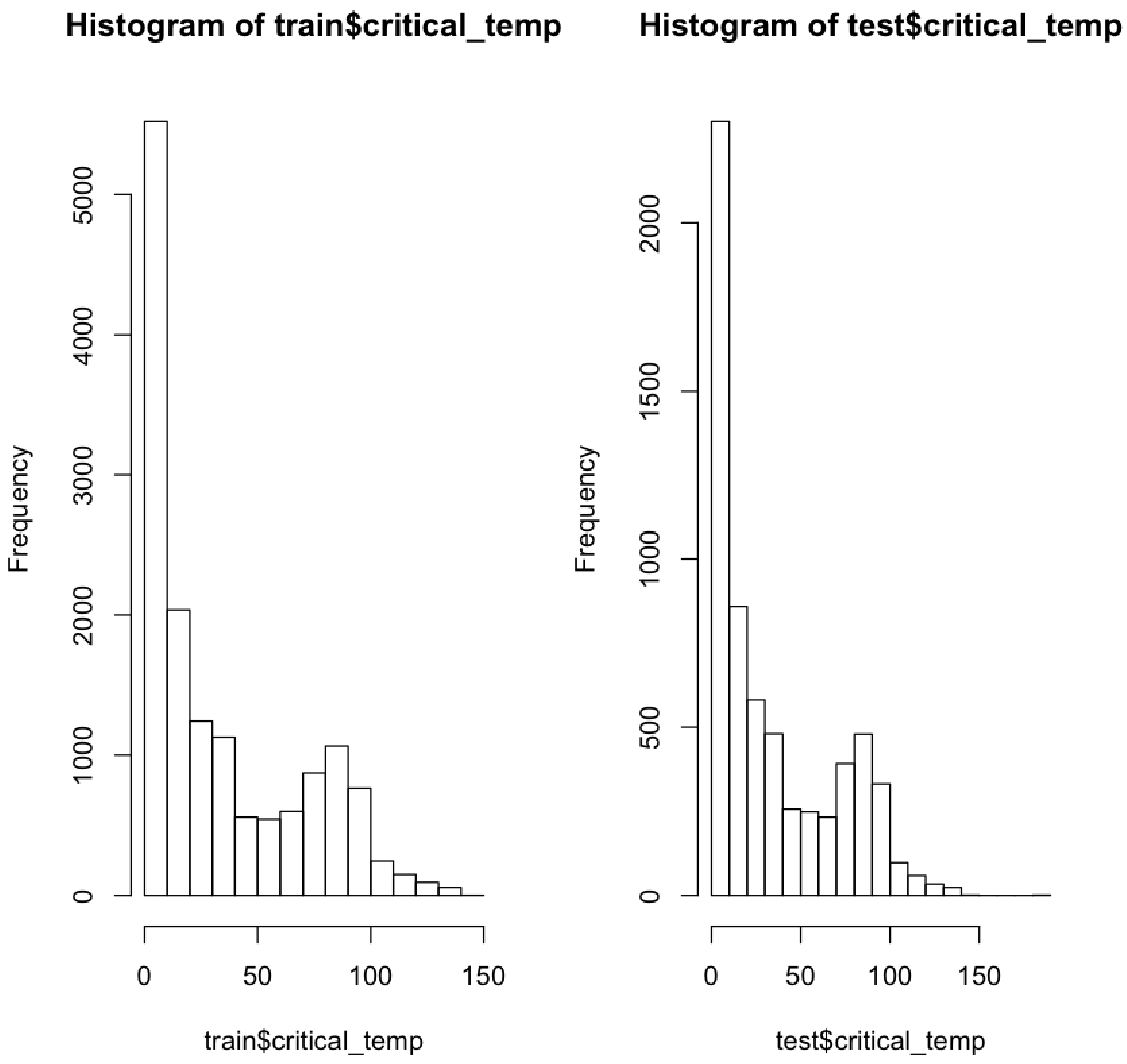
Here, we will split the data based on 70:30 rule, which is 70% of the data for the training set and 30% for the test set. We will use `Random Sampling` method to make this split. This can be achieved by `sample()` function in base of R.

In [6]:
```
## 70% of the sample size
smp_size <- floor(0.70 * nrow(conduct))

## set the seed to make your partition reproducible
set.seed(1237)
train_ind <- sample(seq_len(nrow(conduct)), size = smp_size) # train indices generated by sampling

train <- conduct[train_ind, ]
test <- conduct[-train_ind, ]
```

Now, we will look at the distribution of both `training` and `test` set, to check for any sampling bias introduced while splitting the data. If both the distributions looks similar, we are good to go for our further analysis without introduction of any sampling bias from the `train-test split` .

In [7]:

```
# checking distributions of both samples
par(mfrow=c(1,2))
hist(train$critical_temp)
hist(test$critical_temp)
```



In [8]:

```
# Splitting data from labels
# Training dataset
train.data <- train[,-82]
train.label <- train[,82]

# Testing dataset
test.data <- test[,-82]
test.label <- test[,82]
```

In [9]:
```
# Let's evaluate the correlation matrix
corr <- cor(train)

# Organising the matrix for pairwise correlation of features
corr.m <- data.frame('Var1'=rownames(corr)[row(corr)[upper.tri(corr)]],
            'Var2'=colnames(corr)[col(corr)[upper.tri(corr)]],
            'value'=corr[upper.tri(corr)])

# looking at the correlations
corr.m
```

Var1	Var2	value
number_of_elements	mean_atomic_mass	-0.13983969
number_of_elements	wtd_mean_atomic_mass	-0.35127311
mean_atomic_mass	wtd_mean_atomic_mass	0.81682544
number_of_elements	gmean_atomic_mass	-0.29039543
mean_atomic_mass	gmean_atomic_mass	0.94009473
wtd_mean_atomic_mass	gmean_atomic_mass	0.84841353
number_of_elements	wtd_gmean_atomic_mass	-0.45279942
mean_atomic_mass	wtd_gmean_atomic_mass	0.74761180
wtd_mean_atomic_mass	wtd_gmean_atomic_mass	0.96398655
gmean_atomic_mass	wtd_gmean_atomic_mass	0.85814357
number_of_elements	entropy_atomic_mass	0.93835078
mean_atomic_mass	entropy_atomic_mass	-0.10206879
wtd_mean_atomic_mass	entropy_atomic_mass	-0.30611439
gmean_atomic_mass	entropy_atomic_mass	-0.18670439
wtd_gmean_atomic_mass	entropy_atomic_mass	-0.36812398
number_of_elements	wtd_entropy_atomic_mass	0.88152779
mean_atomic_mass	wtd_entropy_atomic_mass	-0.09455434
wtd_mean_atomic_mass	wtd_entropy_atomic_mass	-0.40918660
gmean_atomic_mass	wtd_entropy_atomic_mass	-0.22758506
wtd_gmean_atomic_mass	wtd_entropy_atomic_mass	-0.48070953
entropy_atomic_mass	wtd_entropy_atomic_mass	0.89016544
number_of_elements	range_atomic_mass	0.68163304
mean_atomic_mass	range_atomic_mass	0.12463575
wtd_mean_atomic_mass	range_atomic_mass	-0.14327306
gmean_atomic_mass	range_atomic_mass	-0.17748233
wtd_gmean_atomic_mass	range_atomic_mass	-0.35145075
entropy_atomic_mass	range_atomic_mass	0.53566087
wtd_entropy_atomic_mass	range_atomic_mass	0.62229841
number_of_elements	wtd_range_atomic_mass	-0.32022824
mean_atomic_mass	wtd_range_atomic_mass	0.44575743
⋮	⋮	⋮
mean_FusionHeat	critical_temp	-0.38188891
wtd_mean_FusionHeat	critical_temp	-0.38999195
gmean_FusionHeat	critical_temp	-0.42780527
wtd_gmean_FusionHeat	critical_temp	-0.42833103
entropy_FusionHeat	critical_temp	0.55585193
wtd_entropy_FusionHeat	critical_temp	0.56579681
range_FusionHeat	critical_temp	-0.13890468
wtd_range_FusionHeat	critical_temp	-0.31005469
std_FusionHeat	critical_temp	-0.19931900
wtd_std_FusionHeat	critical_temp	-0.19324288
mean_ThermalConductivity	critical_temp	0.37956938
wtd_mean_ThermalConductivity	critical_temp	0.38242041
gmean_ThermalConductivity	critical_temp	-0.38331677
wtd_gmean_ThermalConductivity	critical_temp	-0.37103291
entropy_ThermalConductivity	critical_temp	0.09255627
wtd_entropy_ThermalConductivity	critical_temp	-0.11089218
range_ThermalConductivity	critical_temp	0.68893915
wtd_range_ThermalConductivity	critical_temp	0.47289818
std_ThermalConductivity	critical_temp	0.65424831
wtd_std_ThermalConductivity	critical_temp	0.72165813
mean_Valence	critical_temp	-0.59971158
wtd_mean_Valence	critical_temp	-0.63239315
gmean_Valence	critical_temp	-0.57259771
wtd_gmean_Valence	critical_temp	-0.61560226
entropy_Valence	critical_temp	0.60209551
wtd_entropy_Valence	critical_temp	0.59074457
range_Valence	critical_temp	-0.14613817
wtd_range_Valence	critical_temp	-0.43683416
std_Valence	critical_temp	-0.21183696
wtd_std_Valence	critical_temp	-0.30500636

2. Data Exploration

Let's look at the statistics of all variables.

In [10]:

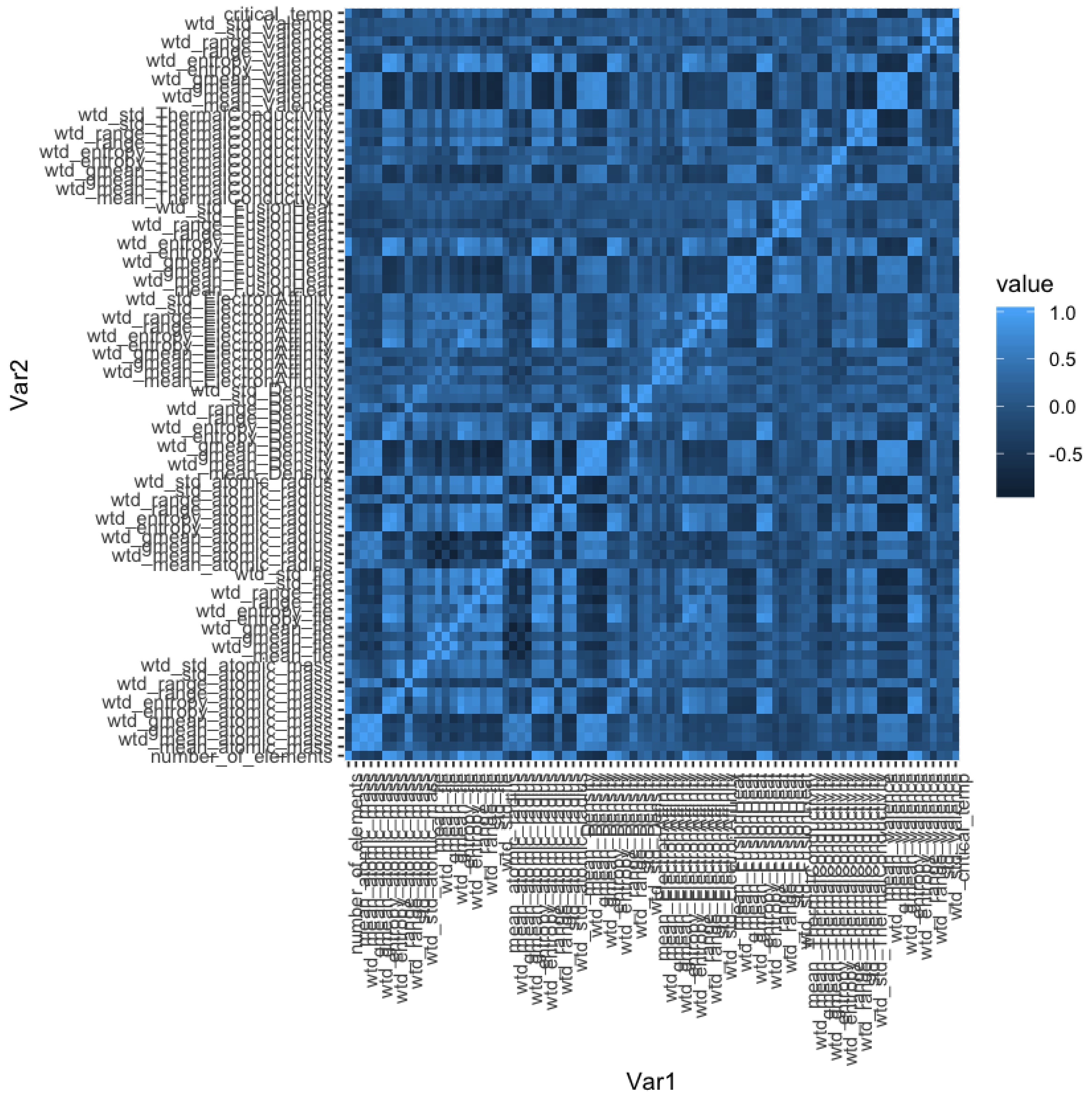
```
summary(conduct)
```

```
number_of_elements mean_atomic_mass  wtd_mean_atomic_mass gmean_atomic_mass
Min.   :1.000      Min.   :  6.941   Min.   :  6.423      Min.   :  5.321
1st Qu.:3.000      1st Qu.: 72.458   1st Qu.: 52.144      1st Qu.: 58.041
Median :4.000      Median : 84.923   Median : 60.697      Median : 66.362
Mean   :4.115      Mean   : 87.558   Mean   : 72.988      Mean   : 71.291
3rd Qu.:5.000      3rd Qu.:100.404   3rd Qu.: 86.104      3rd Qu.: 78.117
Max.   :9.000      Max.   :208.980   Max.   :208.980      Max.   :208.980
wtd_gmean_atomic_mass entropy_atomic_mass wtd_entropy_atomic_mass
Min.   :  1.961       Min.   :0.0000      Min.   :0.0000
1st Qu.: 35.249       1st Qu.:0.9667      1st Qu.:0.7754
Median : 39.918       Median :1.1995      Median :1.1468
Mean   : 58.540       Mean   :1.1656      Mean   :1.0639
3rd Qu.: 73.113       3rd Qu.:1.4445      3rd Qu.:1.3594
Max.   :208.980       Max.   :1.9838      Max.   :1.9582
range_atomic_mass wtd_range_atomic_mass std_atomic_mass  wtd_std_atomic_mass
Min.   :  0.00    Min.   :  0.00        Min.   :  0.00   Min.   :  0.00
1st Qu.: 78.51    1st Qu.: 16.82        1st Qu.: 32.89   1st Qu.: 28.54
Median :122.91    Median : 26.64        Median : 45.12   Median : 44.29
Mean   :115.60    Mean   : 33.23        Mean   : 44.39   Mean   : 41.45
3rd Qu.:154.12    3rd Qu.: 38.36        3rd Qu.: 59.32   3rd Qu.: 53.63
Max.   :207.97    Max.   :205.59        Max.   :101.02   Max.   :101.02
   mean_fie       wtd_mean_fie      gmean_fie      wtd_gmean_fie
Min.   : 375.5   Min.   : 375.5   Min.   : 375.5   Min.   : 375.5
1st Qu.: 723.7   1st Qu.: 738.9   1st Qu.: 692.5   1st Qu.: 720.1
Median : 764.9   Median : 890.0   Median : 728.0   Median : 856.2
Mean   : 769.6   Mean   : 870.4   Mean   : 737.5   Mean   : 832.8
3rd Qu.: 796.3   3rd Qu.:1004.1   3rd Qu.: 765.7   3rd Qu.: 937.6
Max.   :1313.1   Max.   :1348.0   Max.   :1313.1   Max.   :1327.6
 entropy_fie    wtd_entropy_fie    range_fie      wtd_range_fie
Min.   :0.000   Min.   :0.0000   Min.   :   0.0   Min.   :   0.0
1st Qu.:1.086   1st Qu.:0.7538   1st Qu.: 262.4   1st Qu.: 291.1
Median :1.356   Median :0.9168   Median : 764.1   Median : 510.4
Mean   :1.299   Mean   :0.9267   Mean   : 572.2   Mean   : 483.5
3rd Qu.:1.551   3rd Qu.:1.0618   3rd Qu.: 810.6   3rd Qu.: 690.7
Max.   :2.158   Max.   :2.0386   Max.   :1304.5   Max.   :1251.9
   std_fie       wtd_std_fie     mean_atomic_radius wtd_mean_atomic_radius
Min.   :  0.0   Min.   :  0.00   Min.   : 48.0      Min.   : 48.0
1st Qu.:114.1   1st Qu.: 92.99   1st Qu.:149.3      1st Qu.:112.1
Median :266.4   Median :258.45   Median :160.2      Median :126.0
Mean   :215.6   Mean   :224.05   Mean   :158.0      Mean   :134.7
3rd Qu.:297.7   3rd Qu.:342.66   3rd Qu.:169.9      3rd Qu.:158.3
Max.   :499.7   Max.   :479.16   Max.   :298.0      Max.   :298.0
gmean_atomic_radius wtd_gmean_atomic_radius entropy_atomic_radius
Min.   : 48.0       Min.   : 48.00          Min.   :0.000
1st Qu.:133.5       1st Qu.: 89.21          1st Qu.:1.066
Median :142.8       Median :113.18          Median :1.331
Mean   :144.4       Mean   :120.99          Mean   :1.268
3rd Qu.:155.9       3rd Qu.:150.99          3rd Qu.:1.512
Max.   :298.0       Max.   :298.00          Max.   :2.142
wtd_entropy_atomic_radius range_atomic_radius wtd_range_atomic_radius
Min.   :0.0000            Min.   :  0.0       Min.   :  0.00
1st Qu.:0.8522            1st Qu.: 80.0       1st Qu.: 28.60
Median :1.2429            Median :171.0       Median : 43.00
Mean   :1.1311            Mean   :139.3       Mean   : 51.37
3rd Qu.:1.4257            3rd Qu.:205.0       3rd Qu.: 60.22
Max.   :1.9037            Max.   :256.0       Max.   :240.16
std_atomic_radius wtd_std_atomic_radius  mean_Density
Min.   :  0.00    Min.   : 0.00         Min.   :    1.429
1st Qu.: 35.11    1st Qu.:32.02         1st Qu.: 4513.500
Median : 58.66    Median :59.93         Median : 5329.086
Mean   : 51.60    Mean   :52.34         Mean   : 6111.465
3rd Qu.: 69.42    3rd Qu.:73.78         3rd Qu.: 6728.000
Max.   :115.50    Max.   :97.14         Max.   :22590.000
wtd_mean_Density    gmean_Density       wtd_gmean_Density   entropy_Density
Min.   :    1.429   Min.   :    1.429   Min.   :    0.686   Min.   :0.000
1st Qu.: 2999.158   1st Qu.:  883.117   1st Qu.:   66.747   1st Qu.:0.914
Median : 4303.422   Median : 1339.975   Median : 1515.365   Median :1.091
Mean   : 5267.189   Mean   : 3460.692   Mean   : 3117.241   Mean   :1.072
3rd Qu.: 6416.333   3rd Qu.: 5794.965   3rd Qu.: 5766.015   3rd Qu.:1.324
Max.   :22590.000   Max.   :22590.000   Max.   :22590.000   Max.   :1.954
wtd_entropy_Density range_Density   wtd_range_Density  std_Density
Min.   :0.0000      Min.   :    0   Min.   :    0     Min.   :    0
1st Qu.:0.6887      1st Qu.: 6648   1st Qu.: 1657     1st Qu.: 2819
Median :0.8827      Median : 8959   Median : 2083     Median : 3302
Mean   :0.8560      Mean   : 8665   Mean   : 2903     Mean   : 3417
3rd Qu.:1.0809      3rd Qu.: 9779   3rd Qu.: 3409     3rd Qu.: 4004
Max.   :1.7034      Max.   :22589   Max.   :22434     Max.   :10724
wtd_std_Density mean_ElectronAffinity wtd_mean_ElectronAffinity
Min.   :    0   Min.   :  1.50        Min.   :  1.50
1st Qu.: 2564   1st Qu.: 62.09        1st Qu.: 73.35
Median : 3626   Median : 73.10        Median :102.86
Mean   : 3319   Mean   : 76.88        Mean   : 92.72
3rd Qu.: 3959   3rd Qu.: 85.50        3rd Qu.:110.74
Max.   :10411   Max.   :326.10        Max.   :326.10
gmean_ElectronAffinity wtd_gmean_ElectronAffinity entropy_ElectronAffinity
Min.   :  1.50         Min.   :  1.50             Min.   :0.0000
1st Qu.: 33.70         1st Qu.: 50.77             1st Qu.:0.8906
Median : 51.47         Median : 73.17             Median :1.1383
Mean   : 54.36         Mean   : 72.42             Mean   :1.0702
3rd Qu.: 67.51         3rd Qu.: 89.98             3rd Qu.:1.3459
Max.   :326.10         Max.   :326.10             Max.   :1.7677
wtd_entropy_ElectronAffinity range_ElectronAffinity wtd_range_ElectronAffinity
Min.   :0.0000               Min.   :  0.0          Min.   :  0.00
1st Qu.:0.6607               1st Qu.: 86.7          1st Qu.: 34.04
Median :0.7812               Median :127.0          Median : 71.16
Mean   :0.7708               Mean   :120.7          Mean   : 59.33
3rd Qu.:0.8775               3rd Qu.:138.6          3rd Qu.: 76.71
Max.   :1.6754               Max.   :349.0          Max.   :218.70
std_ElectronAffinity wtd_std_ElectronAffinity mean_FusionHeat
Min.   :  0.00       Min.   :  0.00           Min.   :  0.222
1st Qu.: 38.37       1st Qu.: 33.44           1st Qu.:  7.589
Median : 51.13       Median : 48.03           Median :  9.304
Mean   : 48.91       Mean   : 44.41           Mean   : 14.296
3rd Qu.: 56.22       3rd Qu.: 53.32           3rd Qu.: 17.114
Max.   :162.90       Max.   :169.08           Max.   :105.000
wtd_mean_FusionHeat gmean_FusionHeat  wtd_gmean_FusionHeat entropy_FusionHeat
Min.   :  0.222     Min.   :  0.222   Min.   :  0.222      Min.   :0.0000
1st Qu.:  5.033     1st Qu.:  4.110   1st Qu.:  1.322      1st Qu.:0.8333
Median :  8.331     Median :  5.253   Median :  4.930      Median :1.1121
Mean   : 13.848     Mean   : 10.137   Mean   : 10.141      Mean   :1.0933
3rd Qu.: 18.514     3rd Qu.: 13.600   3rd Qu.: 16.429      3rd Qu.:1.3781
Max.   :105.000     Max.   :105.000   Max.   :105.000      Max.   :2.0344
wtd_entropy_FusionHeat range_FusionHeat wtd_range_FusionHeat std_FusionHeat
Min.   :0.0000         Min.   :  0.00   Min.   :  0.000      Min.   : 0.000
1st Qu.:0.6727         1st Qu.: 12.88   1st Qu.:  2.329      1st Qu.: 4.261
Median :0.9950         Median : 12.88   Median :  3.436      Median : 4.948
Mean   :0.9141         Mean   : 21.14   Mean   :  8.219      Mean   : 8.323
3rd Qu.:1.1574         3rd Qu.: 23.20   3rd Qu.: 10.499      3rd Qu.: 9.041
Max.   :1.7472         Max.   :104.78   Max.   :102.675      Max.   :51.635
wtd_std_FusionHeat mean_ThermalConductivity wtd_mean_ThermalConductivity
Min.   : 0.000     Min.   :  0.0266         Min.   :  0.0266
1st Qu.: 4.603     1st Qu.: 61.0000         1st Qu.: 54.1810
Median : 5.501     Median : 96.5044         Median : 73.3333
Mean   : 7.718     Mean   : 89.7069         Mean   : 81.5491
3rd Qu.: 8.018     3rd Qu.:111.0053         3rd Qu.: 99.0629
Max.   :51.680     Max.   :332.5000         Max.   :406.9600
gmean_ThermalConductivity wtd_gmean_ThermalConductivity
```

```
 Min.   :  0.0266          Min.   :  0.023  
 1st Qu.:  8.3398          1st Qu.:  1.087  
 Median : 14.2876          Median :  6.096  
 Mean   : 29.8417          Mean   : 27.308  
 3rd Qu.: 42.3713          3rd Qu.: 47.308  
 Max.   :317.8836          Max.   :376.033  
 entropy_ThermalConductivity wtd_entropy_ThermalConductivity
 Min.   :0.0000              Min.   :0.0000                 
 1st Qu.:0.4578              1st Qu.:0.2507                 
 Median :0.7387              Median :0.5458                 
 Mean   :0.7276              Mean   :0.5400                 
 3rd Qu.:0.9622              3rd Qu.:0.7774                 
 Max.   :1.6340              Max.   :1.6130                 
 range_ThermalConductivity wtd_range_ThermalConductivity
 Min.   :  0.00            Min.   :  0.00               
 1st Qu.: 86.38            1st Qu.: 29.35               
 Median :399.80            Median : 56.56               
 Mean   :250.89            Mean   : 62.03               
 3rd Qu.:399.97            3rd Qu.: 91.87               
 Max.   :429.97            Max.   :401.44               
 std_ThermalConductivity wtd_std_ThermalConductivity  mean_Valence  
 Min.   :  0.00          Min.   :  0.00              Min.   :1.000  
 1st Qu.: 37.93          1st Qu.: 31.99              1st Qu.:2.333  
 Median :135.76          Median :113.56              Median :2.833  
 Mean   : 98.94          Mean   : 96.23              Mean   :3.198  
 3rd Qu.:153.81          3rd Qu.:162.71              3rd Qu.:4.000  
 Max.   :214.99          Max.   :213.30              Max.   :7.000  
 wtd_mean_Valence gmean_Valence   wtd_gmean_Valence entropy_Valence
 Min.   :1.000    Min.   :1.000   Min.   :1.000     Min.   :0.000  
 1st Qu.:2.117    1st Qu.:2.280   1st Qu.:2.091     1st Qu.:1.061  
 Median :2.618    Median :2.615   Median :2.434     Median :1.369  
 Mean   :3.153    Mean   :3.057   Mean   :3.056     Mean   :1.296  
 3rd Qu.:4.026    3rd Qu.:3.728   3rd Qu.:3.915     3rd Qu.:1.589  
 Max.   :7.000    Max.   :7.000   Max.   :7.000     Max.   :2.142  
 wtd_entropy_Valence range_Valence   wtd_range_Valence  std_Valence    
 Min.   :0.0000      Min.   :0.000   Min.   :0.0000    Min.   :0.0000  
 1st Qu.:0.7757      1st Qu.:1.000   1st Qu.:0.9215    1st Qu.:0.4518  
 Median :1.1665      Median :2.000   Median :1.0631    Median :0.8000  
 Mean   :1.0528      Mean   :2.041   Mean   :1.4830    Mean   :0.8393  
 3rd Qu.:1.3308      3rd Qu.:3.000   3rd Qu.:1.9184    3rd Qu.:1.2000  
 Max.   :1.9497      Max.   :6.000   Max.   :6.9922    Max.   :3.0000  
 wtd_std_Valence  critical_temp     
 Min.   :0.0000   Min.   :  0.00021  
 1st Qu.:0.3069   1st Qu.:  5.36500  
 Median :0.5000   Median : 20.00000  
 Mean   :0.6740   Mean   : 34.42122  
 3rd Qu.:1.0204   3rd Qu.: 63.00000  
 Max.   :3.0000   Max.   :185.00000  
```

In [11]:

```
corr.melt <- melt(corr)
ggplot(data = corr.melt, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() + theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

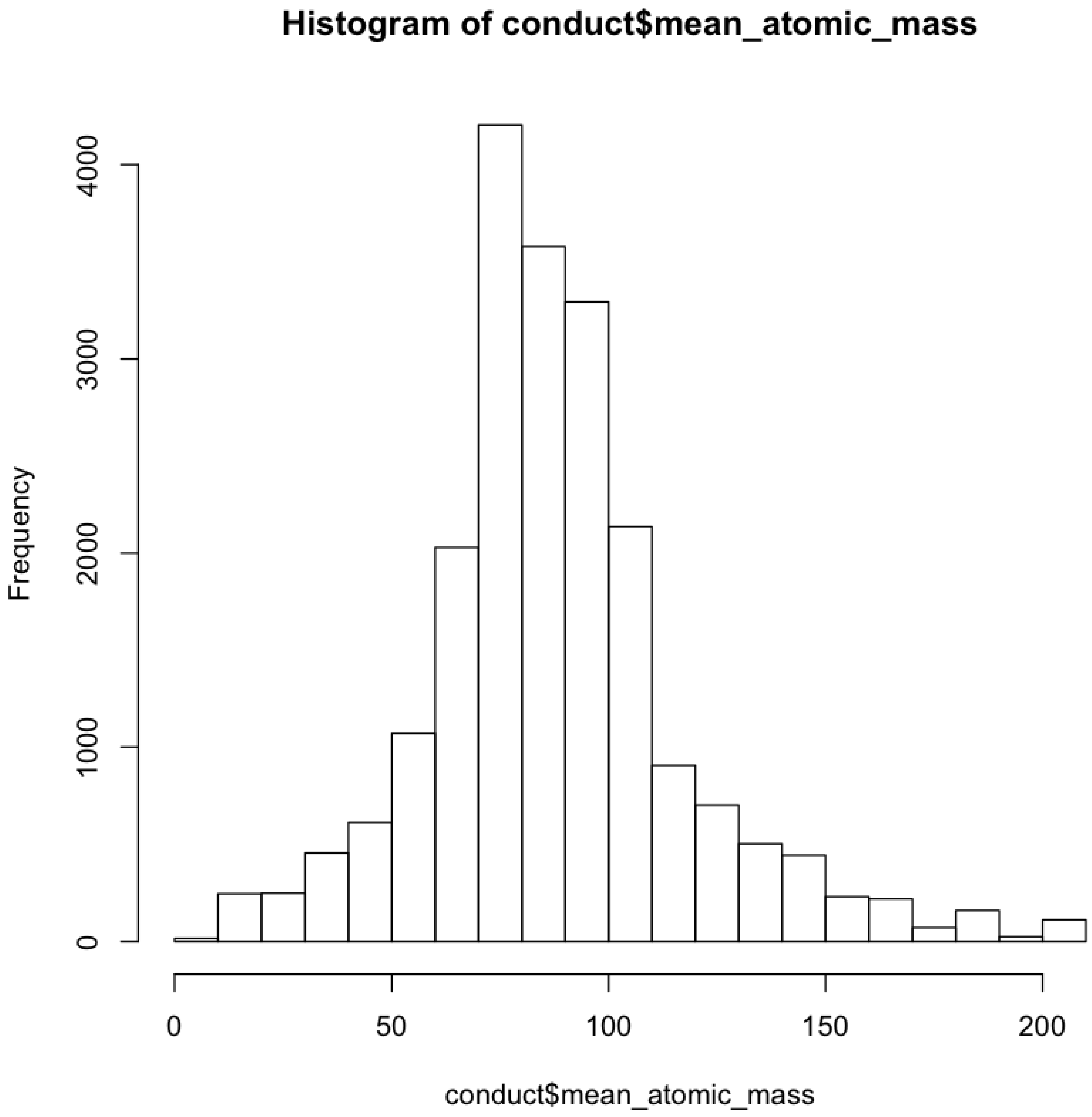


Not much insights can be inferred from this plot, since there's so many features to visualise. We will devide the dataset, and visualise all the properties individually for the ease of perception.

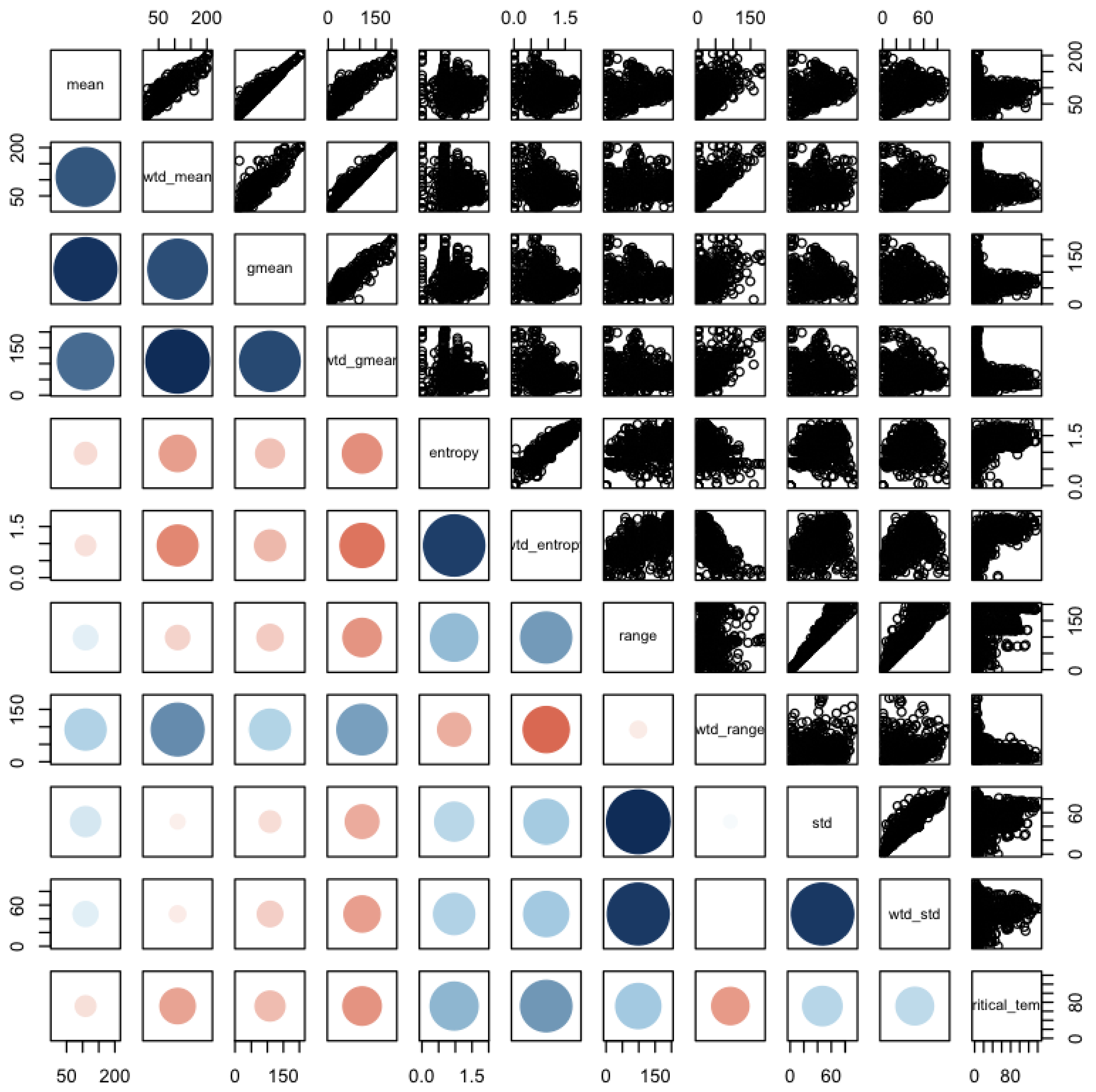
In [12]:
```
colorRange <- c('#69091e', '#e37f65', 'white', '#aed2e6', '#042f60')
## colorRamp() returns a function which takes as an argument a number
## on [0,1] and returns a color in the gradient in colorRange
myColorRampFunc <- colorRamp(colorRange)

panel.cor <- function(w, z, ...) {
  correlation <- cor(w, z)

  ## because the func needs [0,1] and cor gives [-1,1], we need to shift and scale it
  col <- rgb(myColorRampFunc((1 + correlation) / 2 ) / 255 )

  ## square it to avoid visual bias due to "area vs diameter"
  radius <- sqrt(abs(correlation))
  radians <- seq(0, 2*pi, len = 50) # 50 is arbitrary
  x <- radius * cos(radians)
  y <- radius * sin(radians)
  ## make them full loops
  x <- c(x, tail(x,n=1))
  y <- c(y, tail(y,n=1))

  ## trick: "don't create a new plot" thing by following the
  ## advice here: http://www.r-bloggers.com/multiple-y-axis-in-a-r-plot/
  ## This allows
  par(new=TRUE)
  plot(0, type='n', xlim=c(-1,1), ylim=c(-1,1), axes=FALSE, asp=1)
  polygon(x, y, border=col, col=col)
}

# Following function accepts the start and end, and returns the sliced data based on those
filtered_data <- function(data, start, end){
    plot_data <- data[,c(names(data)[start:end],names(data)[82])]
    # setting names of columns for easy identification of characterstic of selected property.
    colnames(plot_data) <-  c('mean','wtd_mean','gmean','wtd_gmean','entropy','wtd_entropy','range','wtd_range','std','wtd_std','critical_temp')
    return(plot_data)
}


# filtering data for all properties
property1 <- filtered_data(conduct,2,11)
property2 <-filtered_data(conduct,12,21)
property3 <-filtered_data(conduct,22,31)
property4 <-filtered_data(conduct,32,41)
property5 <-filtered_data(conduct,42,51)
property6 <-filtered_data(conduct,52,61)
property7 <-filtered_data(conduct,62,71)
property8 <-filtered_data(conduct,72,81)
```

Atomic Mass

In [13]:
```
hist(conduct$mean_atomic_mass)
```



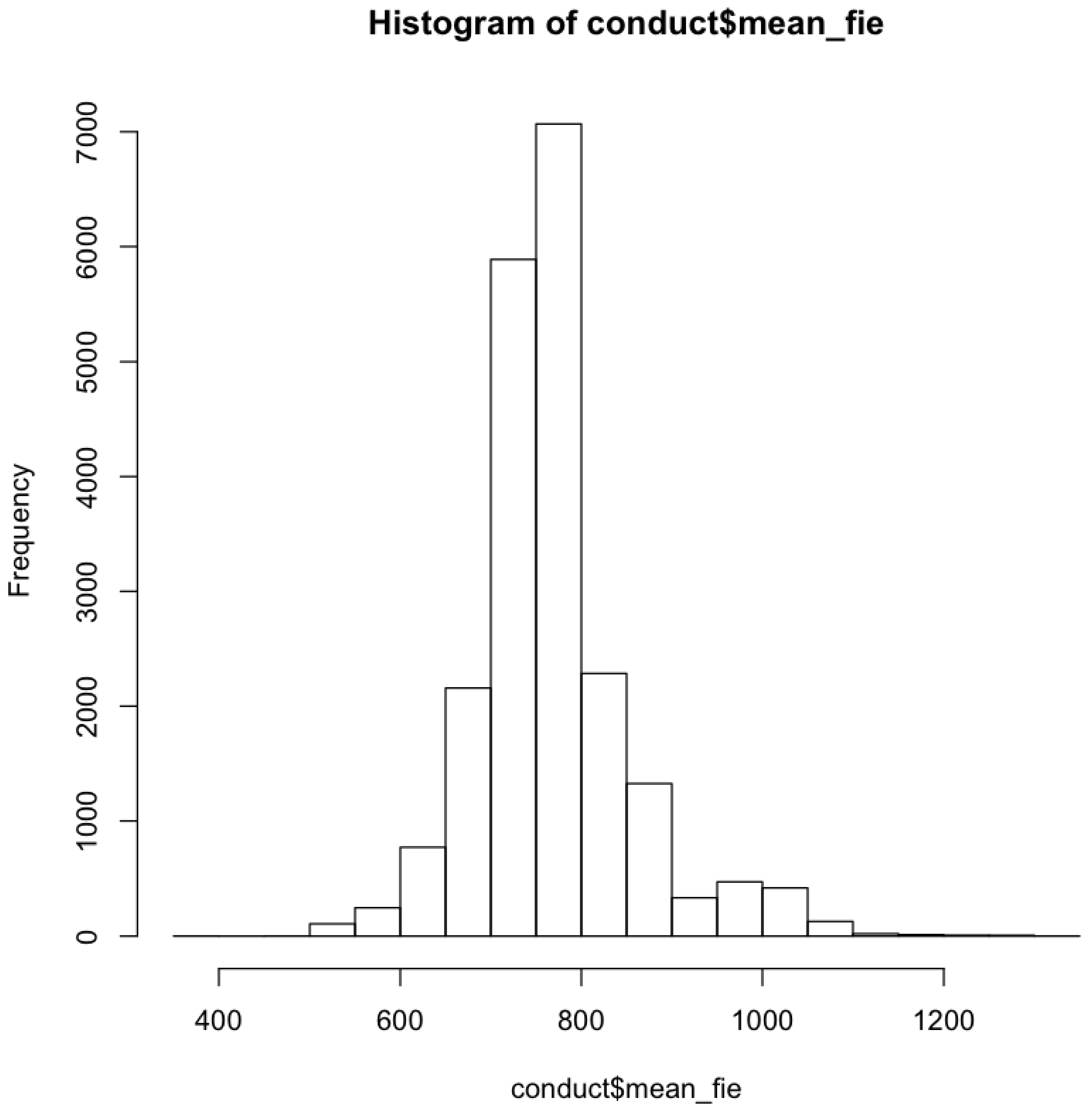
```
In [14]: pairs(property1[sample.int(nrow(property1),1000),], lower.panel=panel.cor)
```



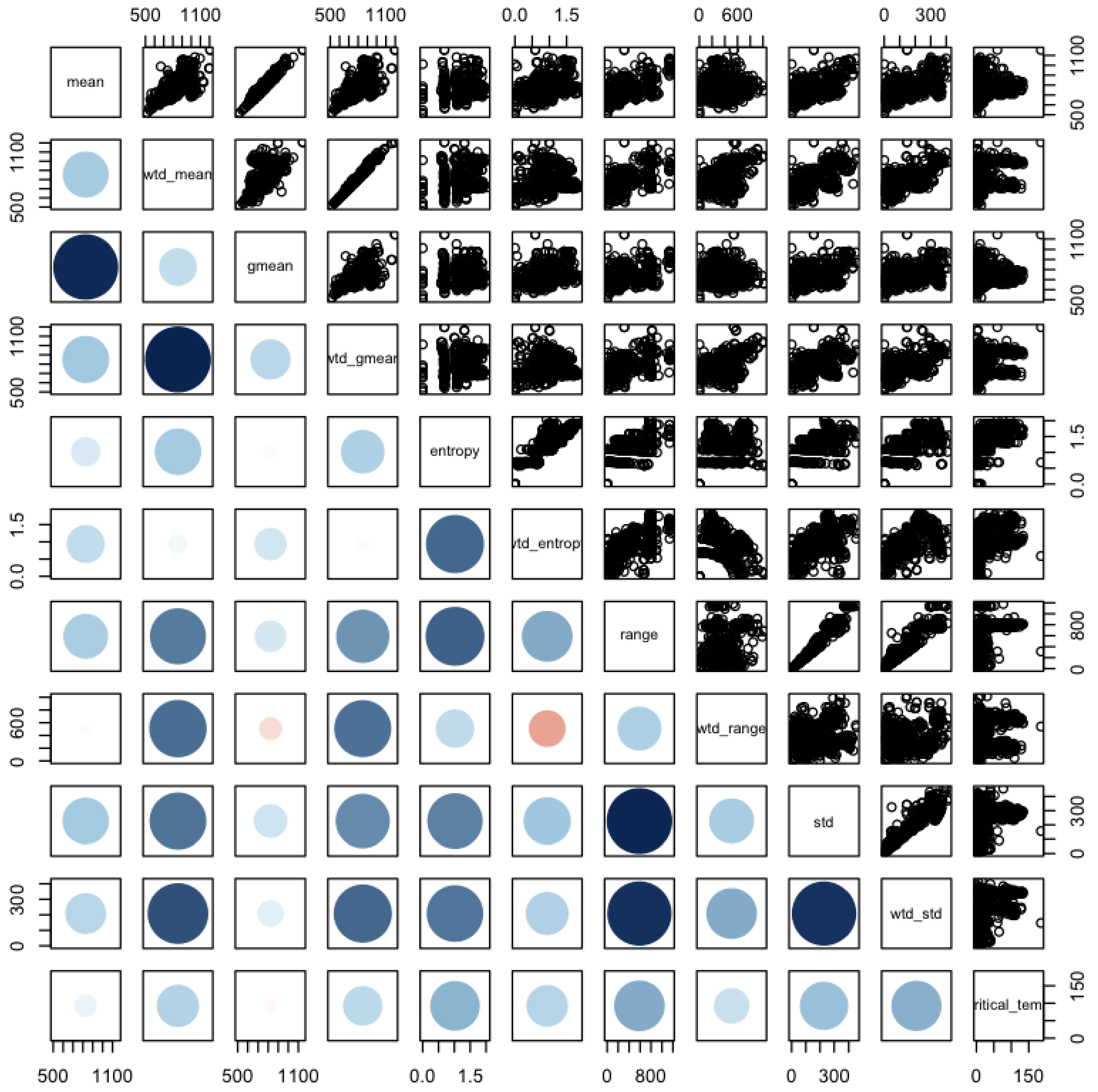
- The distribution of average values of atomic mass is normally distributed.
- `etropy` and `wtd_entropy` are highly positively corelated to `critical temperature`.
- `wtd_mean` and `wtd_gmean` are highly negatively correlated with the `critical temperature`.
- `entropy` and `wtd_entropy` shows some non-colinear relationship with `critical temperature`.

First Ionization Energy

```
In [15]: hist(conduct$mean_fie)
```



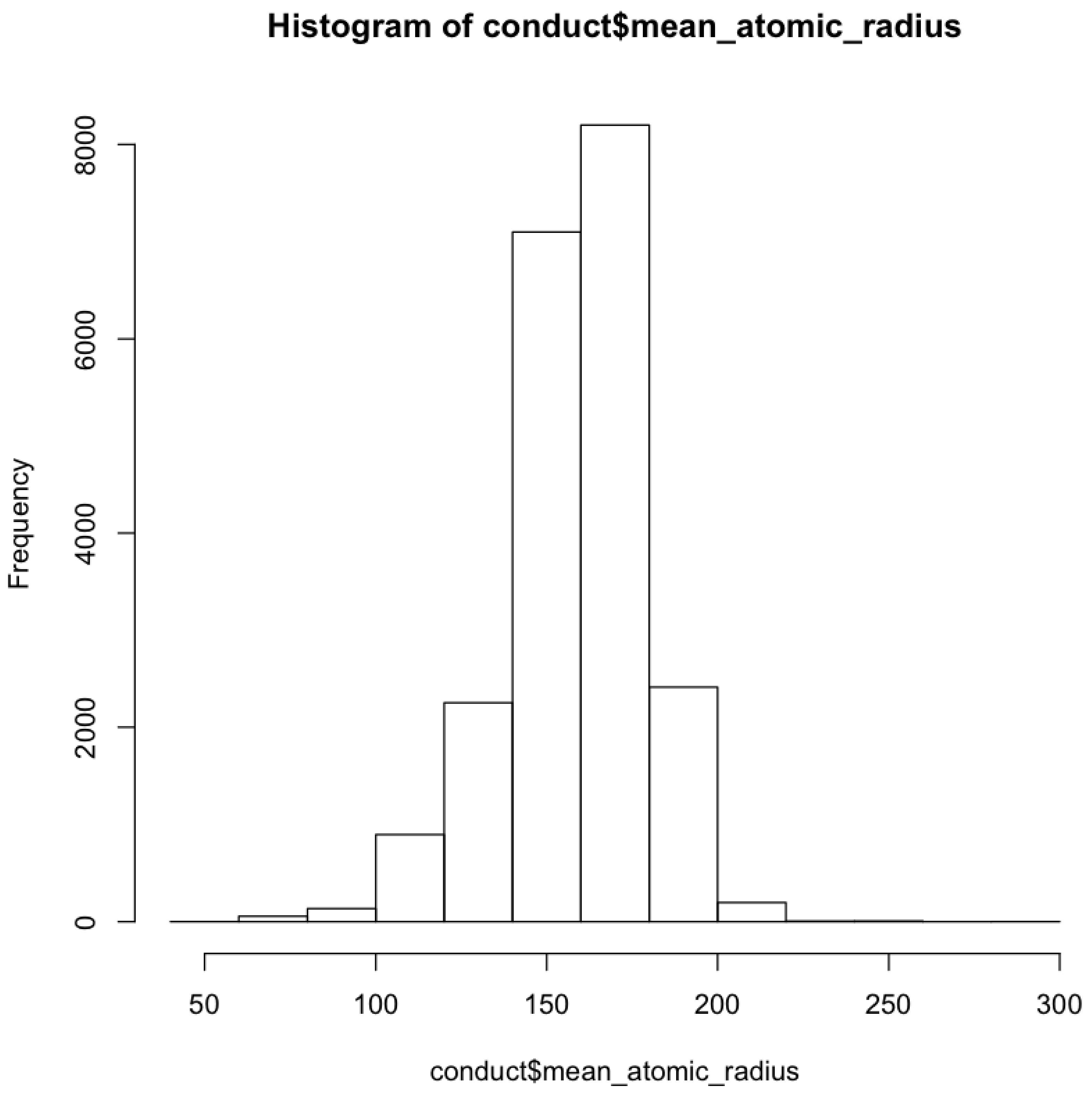
```
In [16]: pairs(property2[sample.int(nrow(property2),1000),], lower.panel=panel.cor)
```



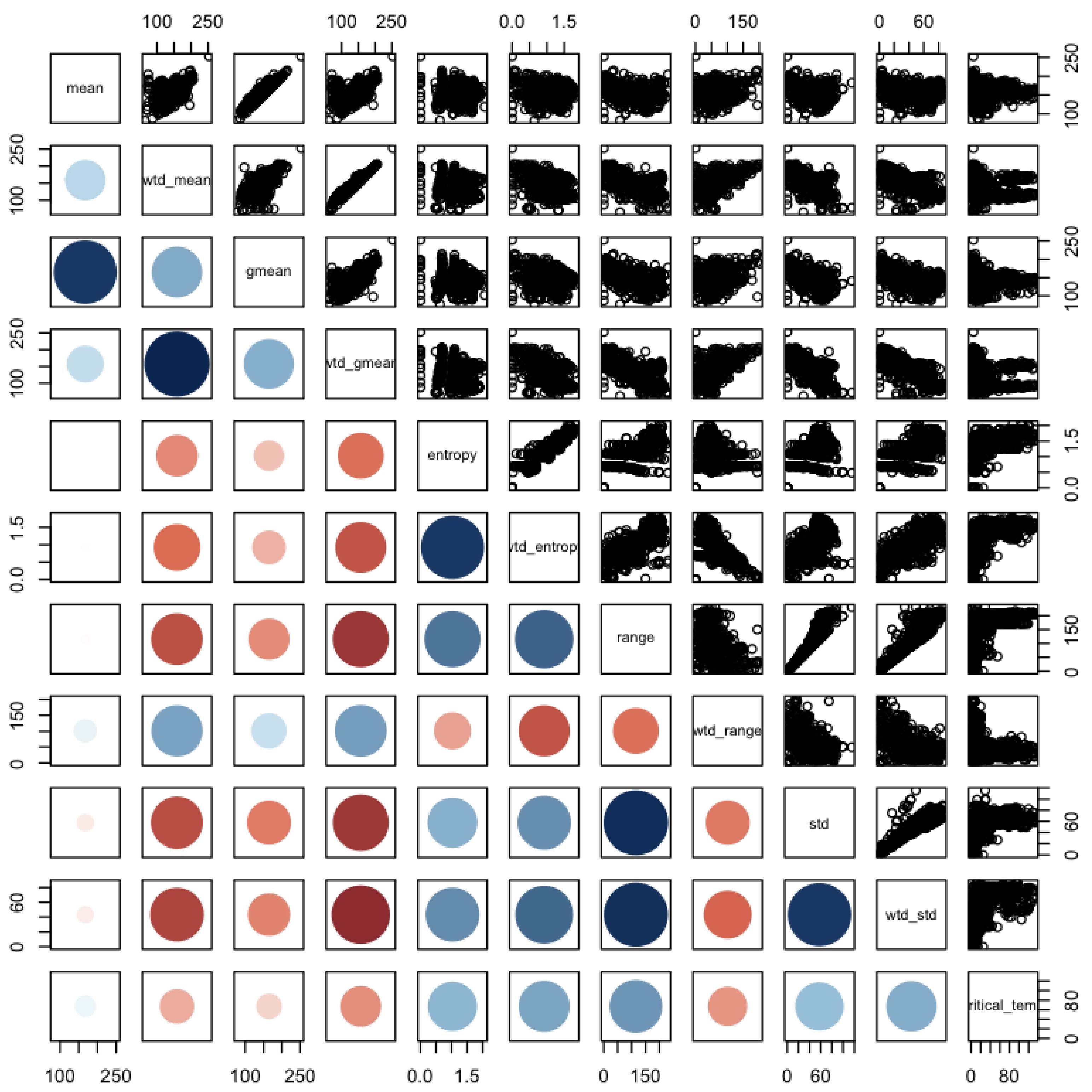
- The distribution of average values of First Ionization Energy is normally distributed.
- `etropy` , `wtd_std` and `range` high correlation with `critical temperature` .
- `entropy` and `wtd_entropy` shows some non-colinear relationship with `critical temperature` .

Atomic Radius

```
In [17]: hist(conduct$mean_atomic_radius)
```



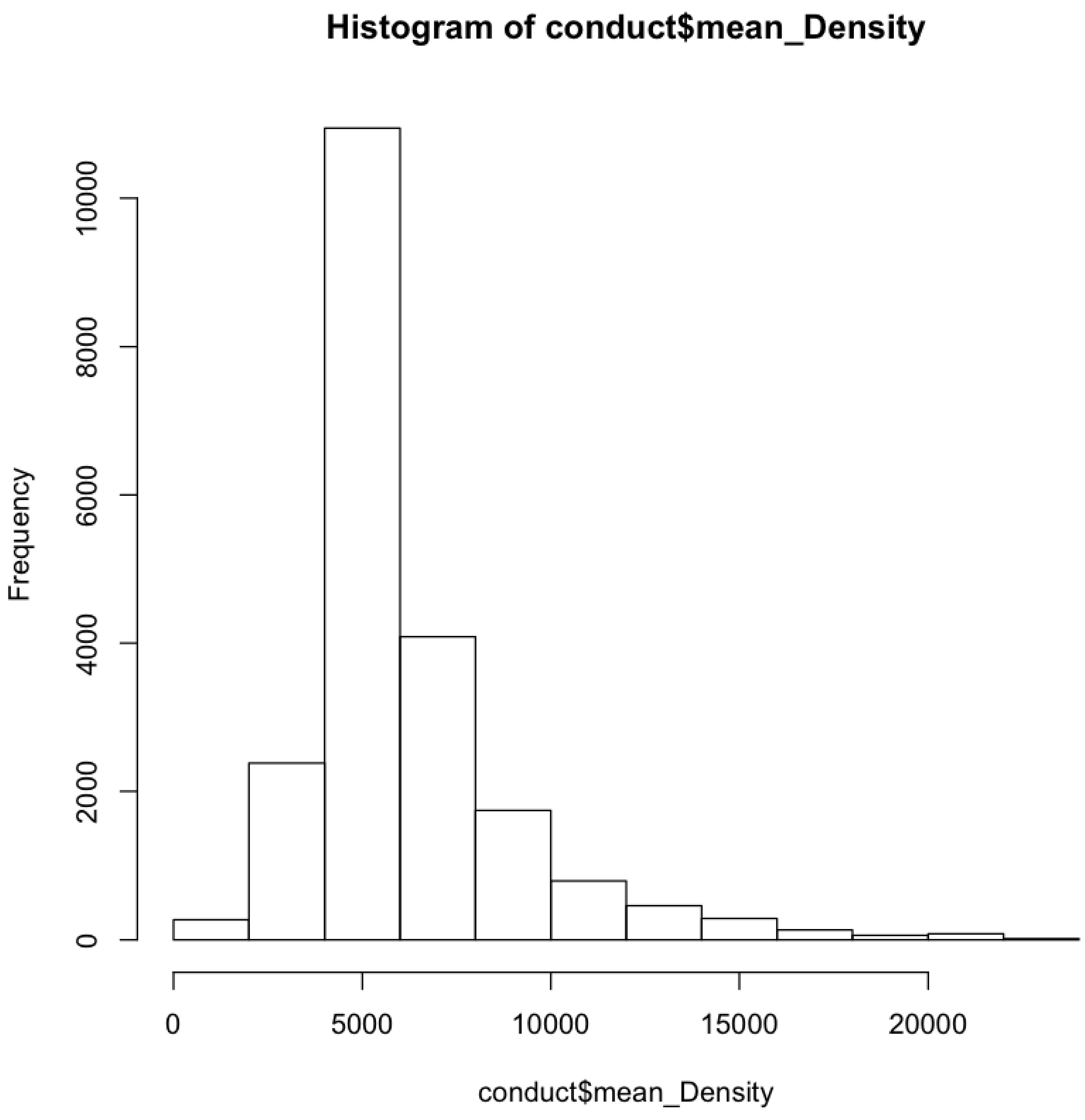
```
In [18]: pairs(property3[sample.int(nrow(property3),1000),], lower.panel=panel.cor)
```



- The distribution of average values of atomic radius is left skewed.
- `range` is highly positively corelated to `critical temperature`.
- `wtd_range` and `wtd_gmean` are highly negatively correlated with the `critical temperature`.
- `entropy` and `wtd_entropy` shows some non-colinear relationship with `critical temperature`.

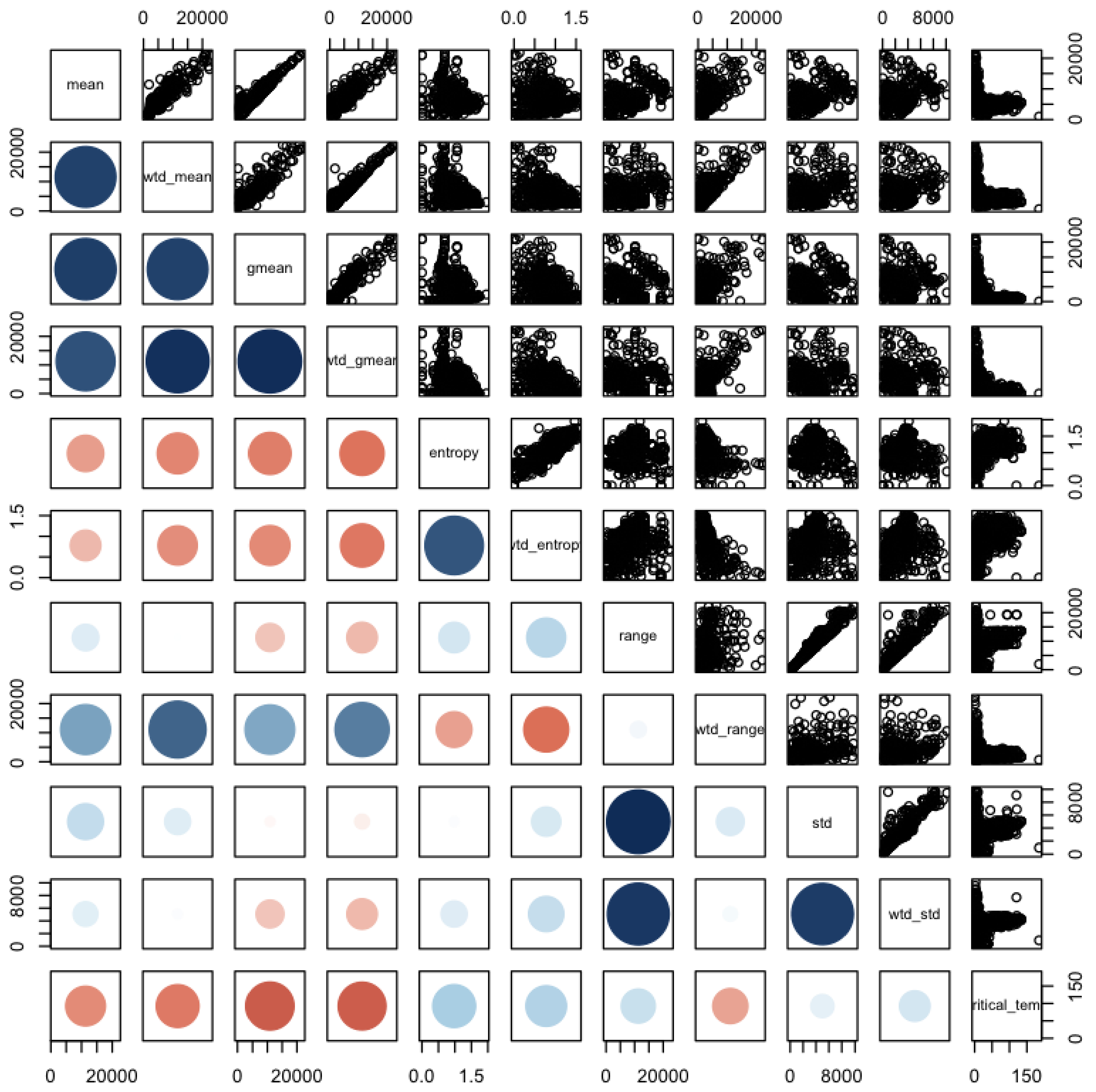
Density

```
In [19]: hist(conduct$mean_Density)
```



In [20]:

```
pairs(property4[sample.int(nrow(property4),1000),], lower.panel=panel.cor)
```

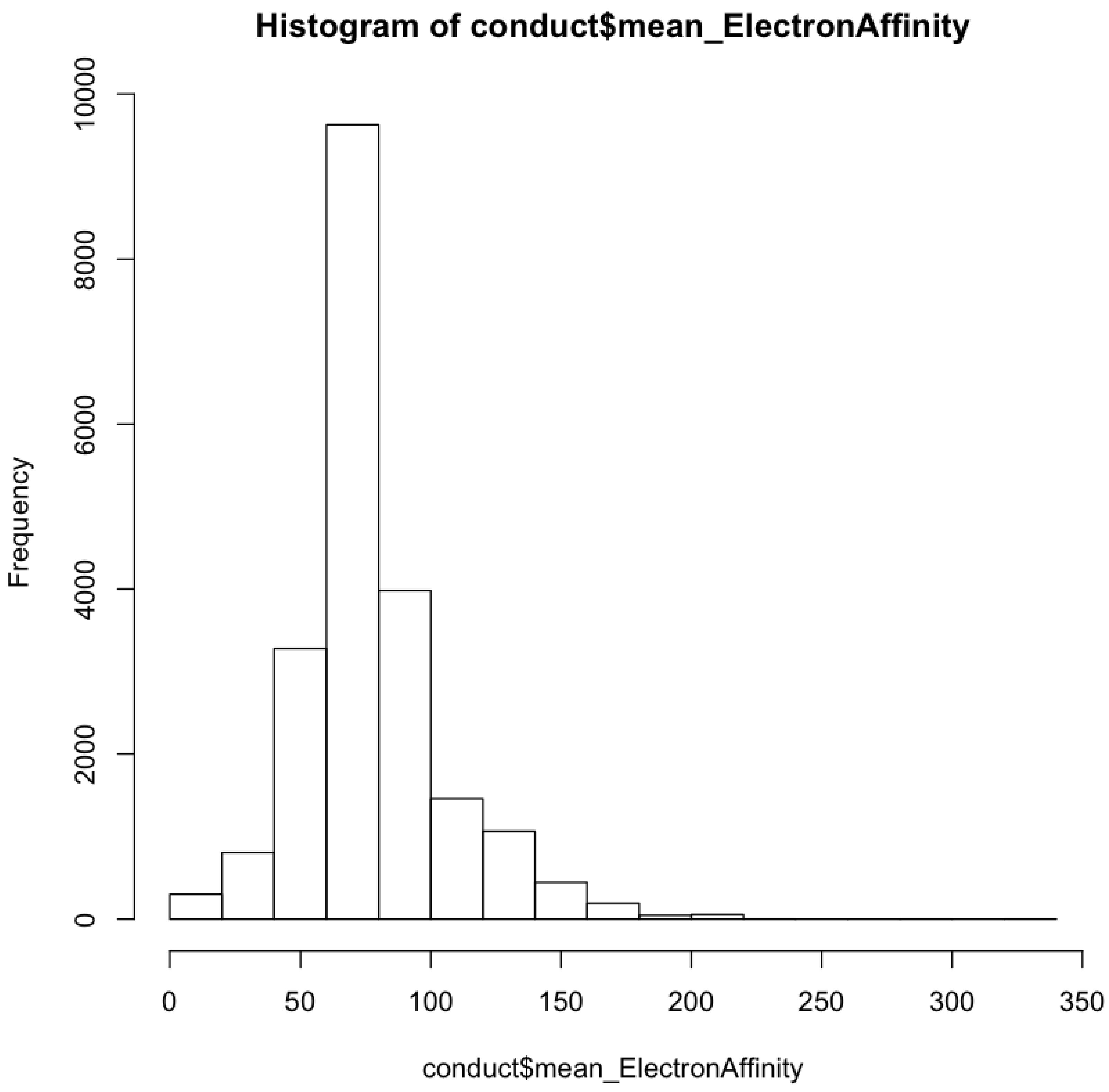


- The distribution of average values of Density is right skewed.
- `wtd_mean` and `wtd_gmean` are highly negatively correlated with the `critical temperature`.
- `entropy` and `wtd_entropy` shows some non-colinear relationship with `critical temperature`.

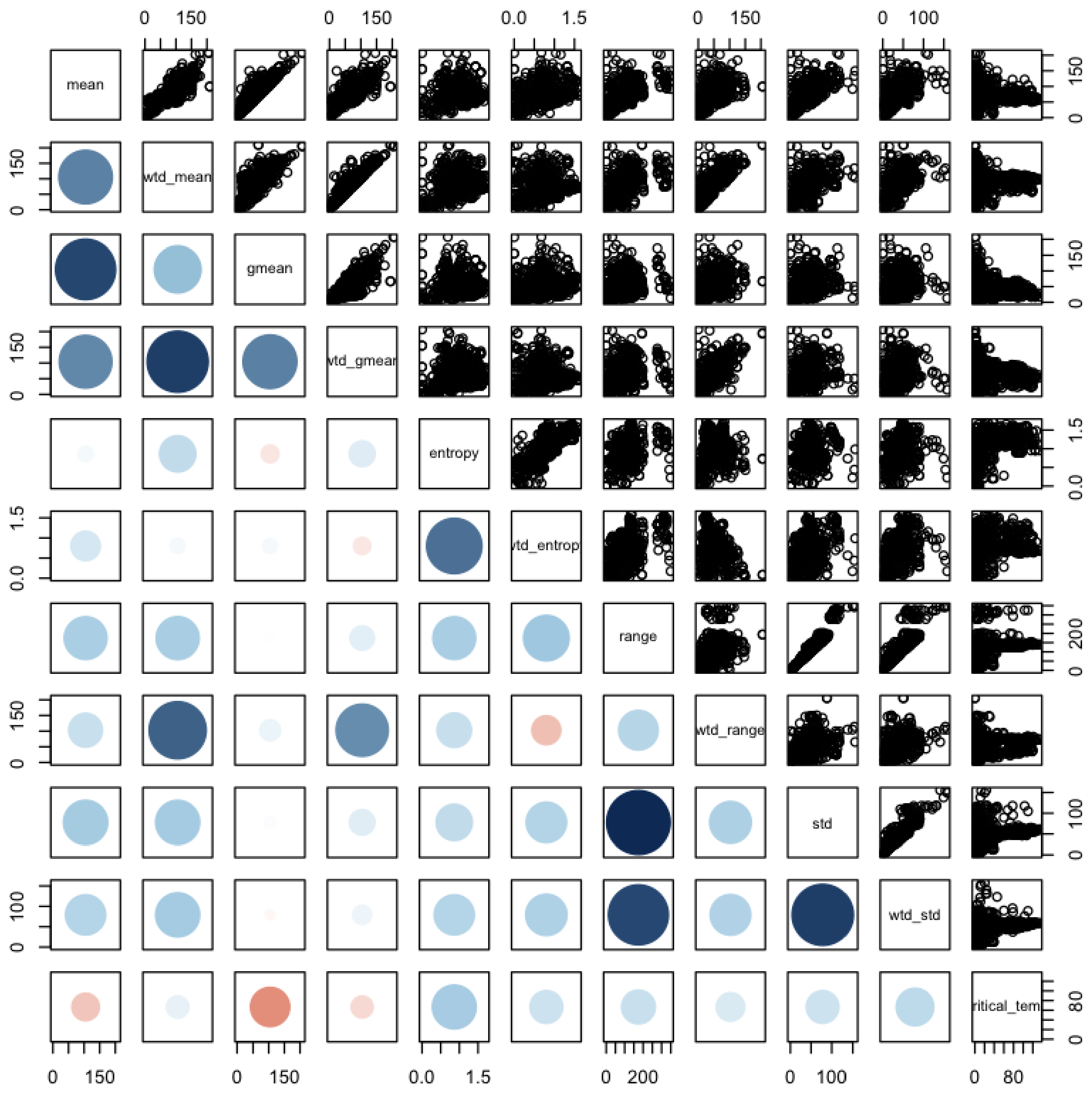
Electron Affinity

In [21]:

```
hist(conduct$mean_ElectronAffinity)
```



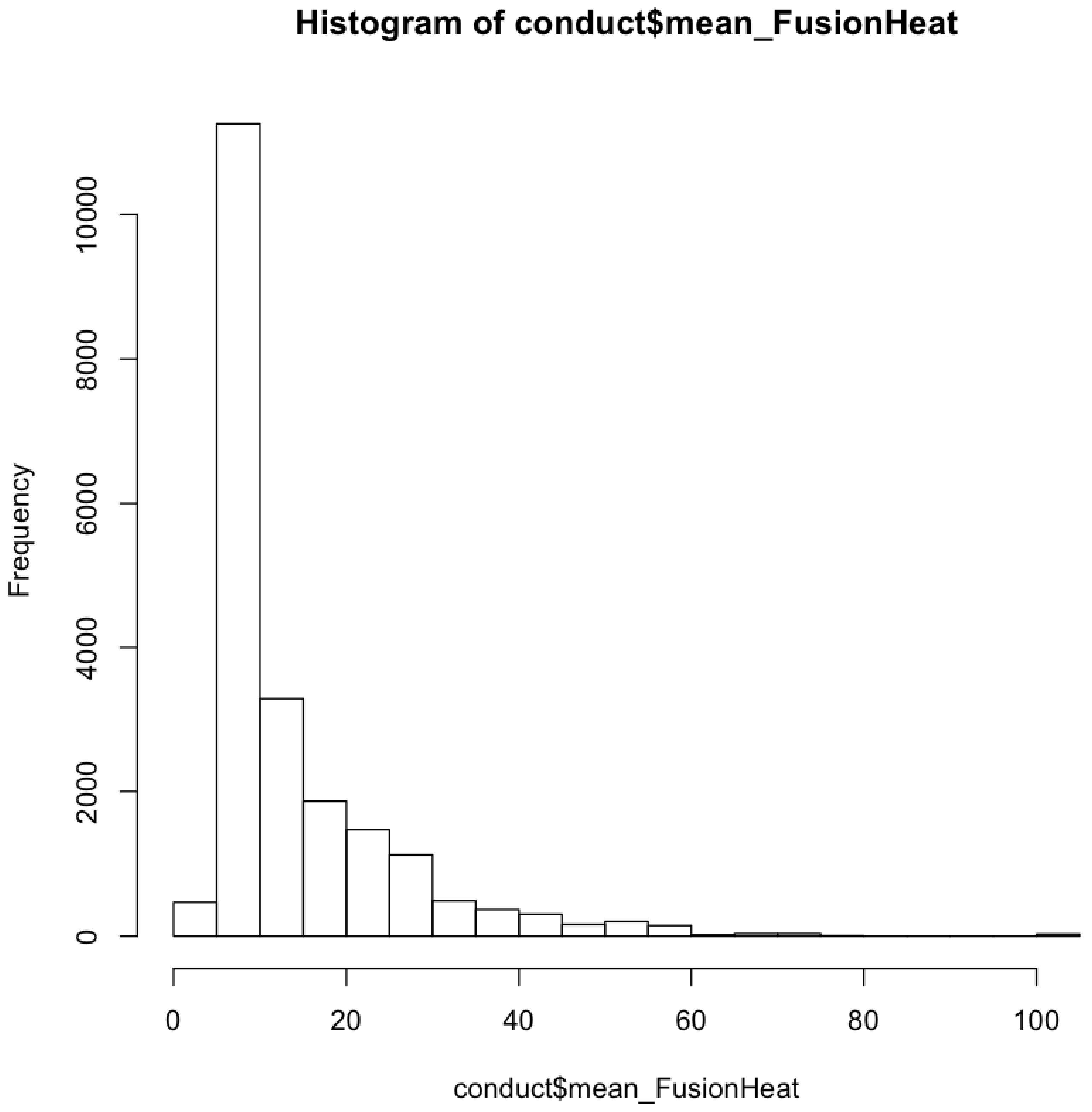
```
In [22]: pairs(property5[sample.int(nrow(property5),1000),], lower.panel=panel.cor)
```



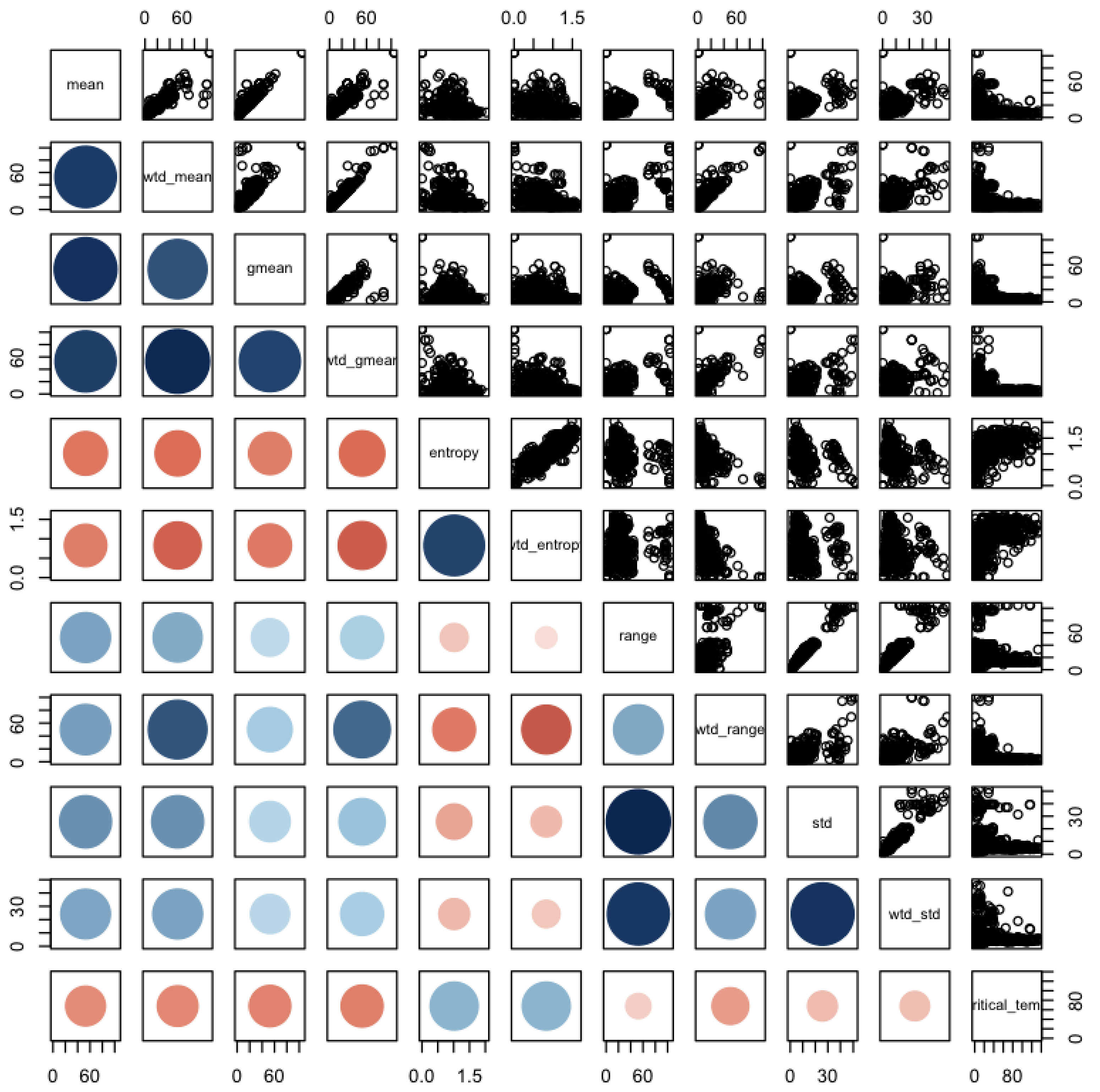
- The distribution of average values of Electron Affinity is normally distributed.
- `etropy` is positively corelated to `critical temperature`.
- `gmean` is negatively correlated with the `critical temperature`.
- `entropy` and `wtd_entropy` shows some non-colinear relationship with `critical temperature`.

Fusion Heat

```
In [23]: hist(conduct$mean_FusionHeat)
```



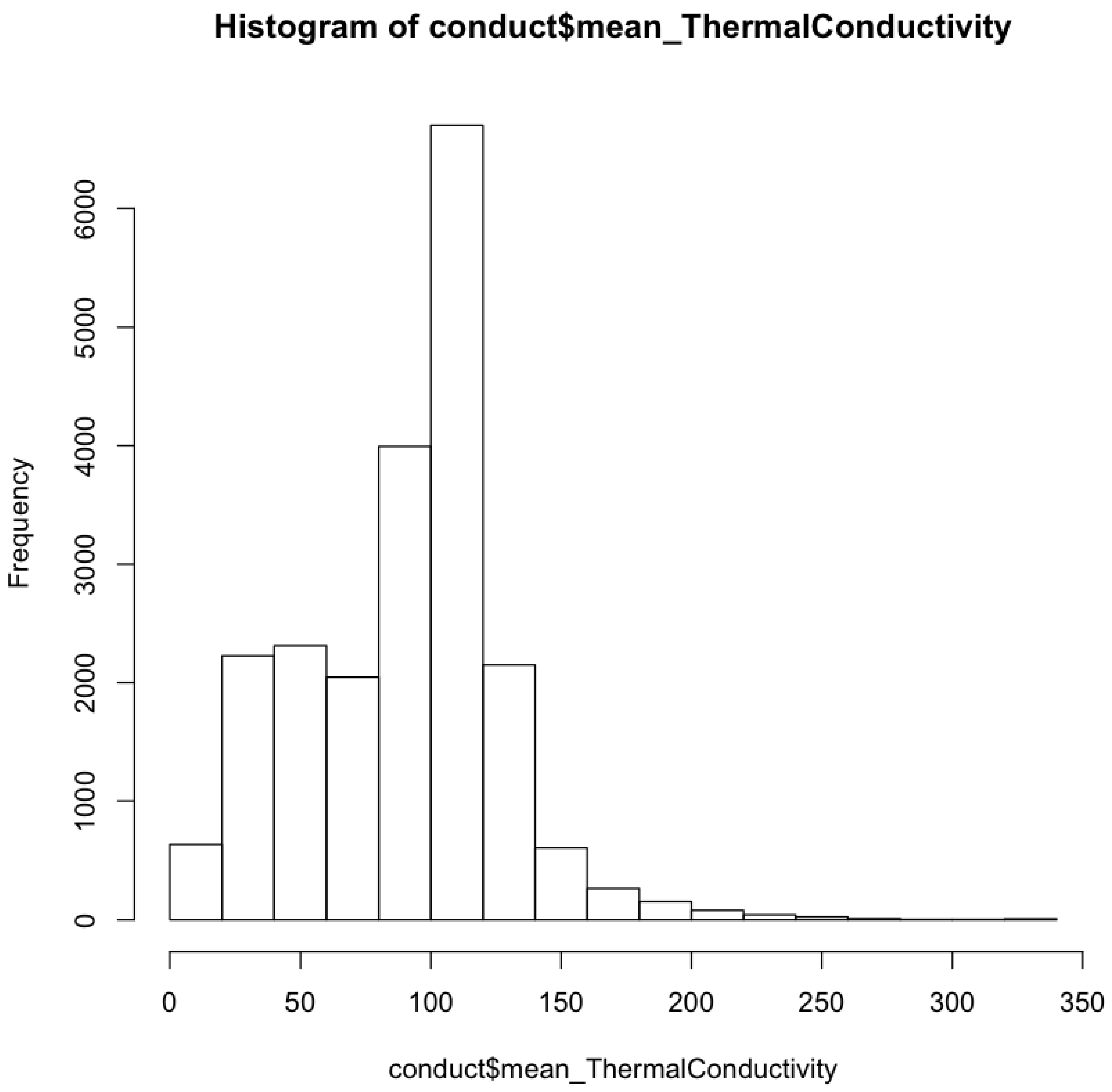
```
In [24]: pairs(property6[sample.int(nrow(property6),1000),], lower.panel=panel.cor)
```



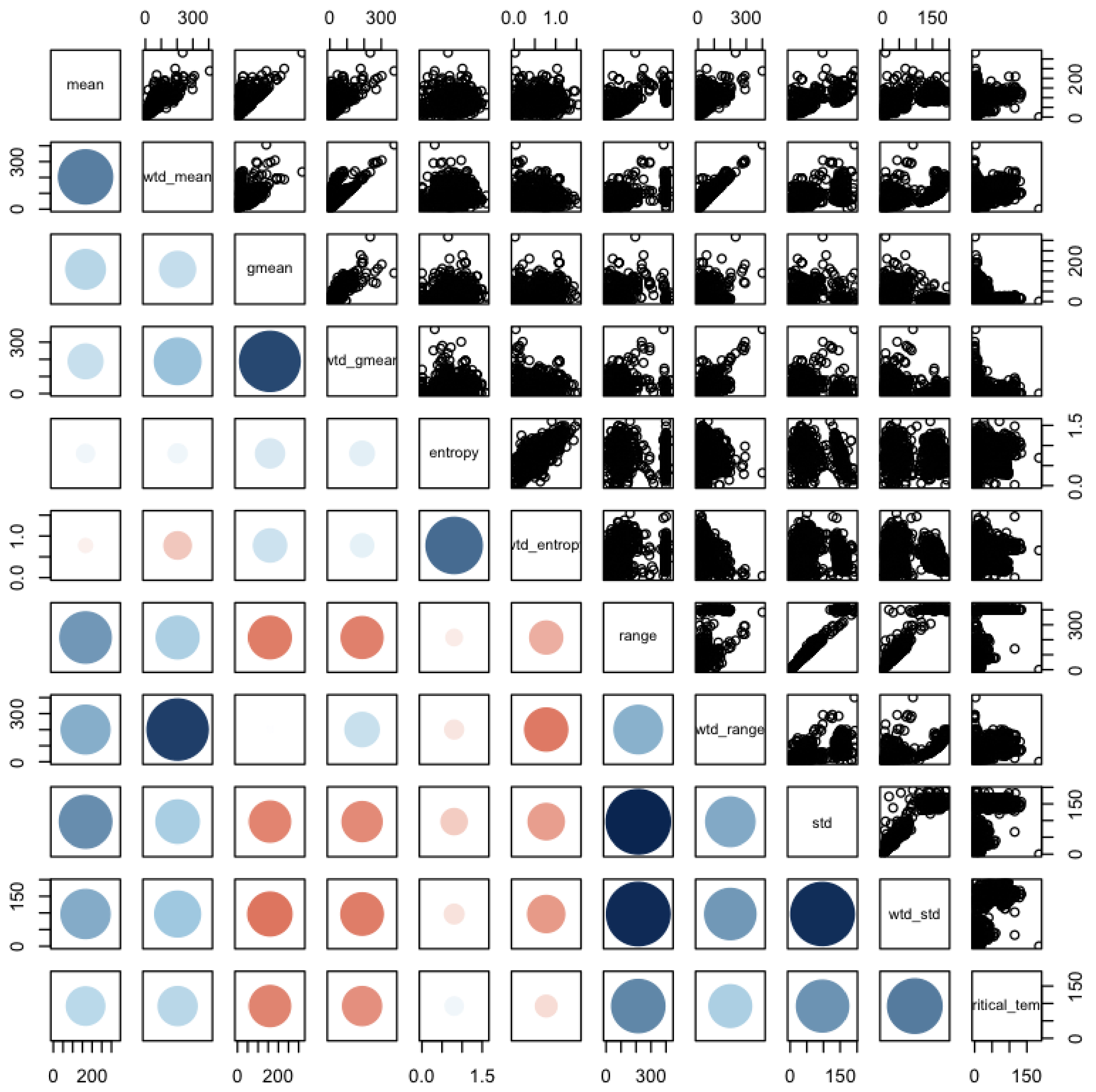
- The distribution of average values of `Fusion Heat` is right skewed.
- `etropy` and `wtd_entropy` are highly positively corelated to `critical temperature`.
- `wtd_mean` and `wtd_gmean` are highly negatively corelated with the `critical temperature`.
- `entropy` and `wtd_entropy` shows some non-colinear relationship with `critical temperature`.

Thermal Conductivity

```
In [25]: hist(conduct$mean_ThermalConductivity)
```



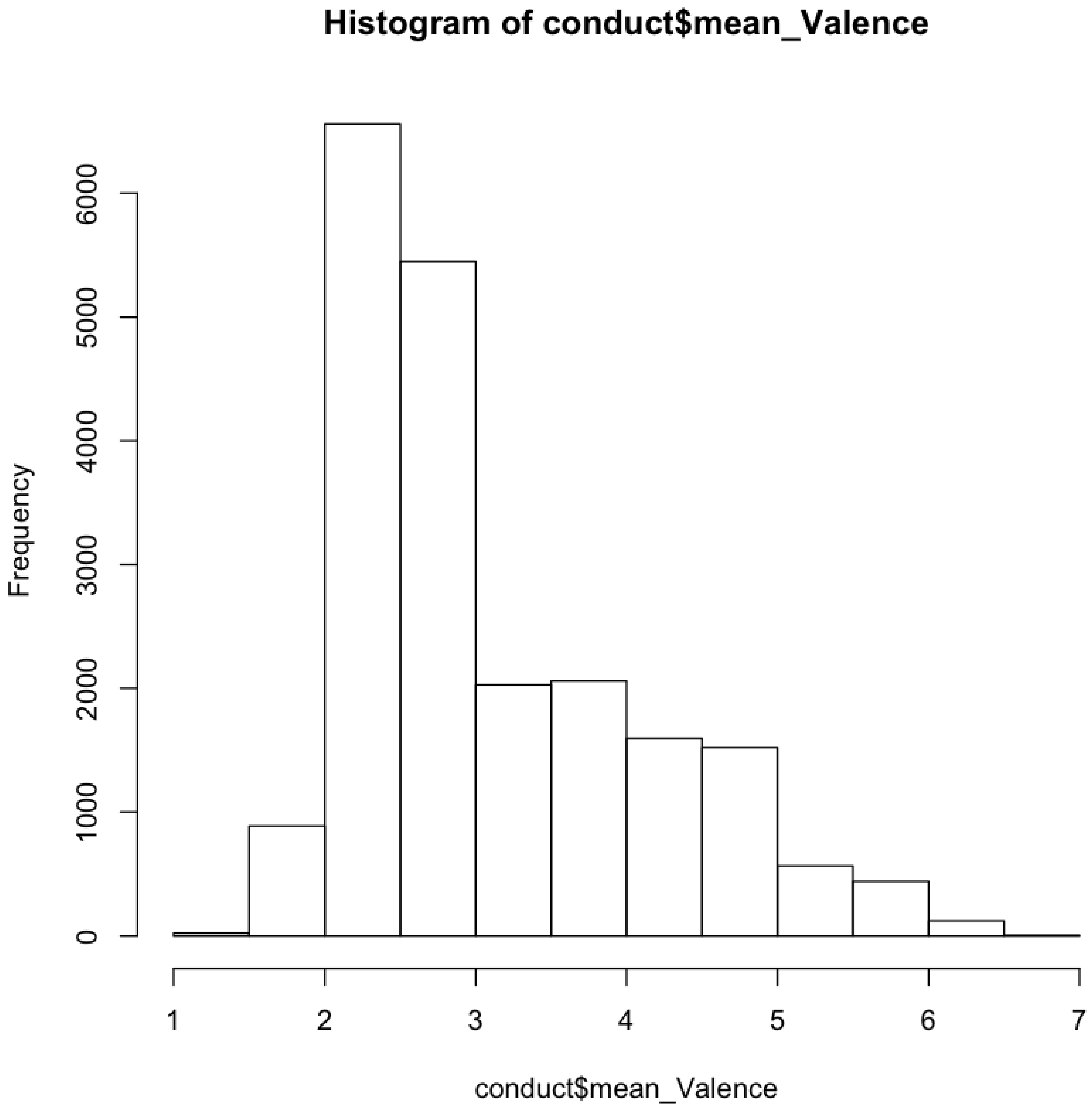
```
In [26]: pairs(property7[sample.int(nrow(property7),1000),], lower.panel=panel.cor)
```



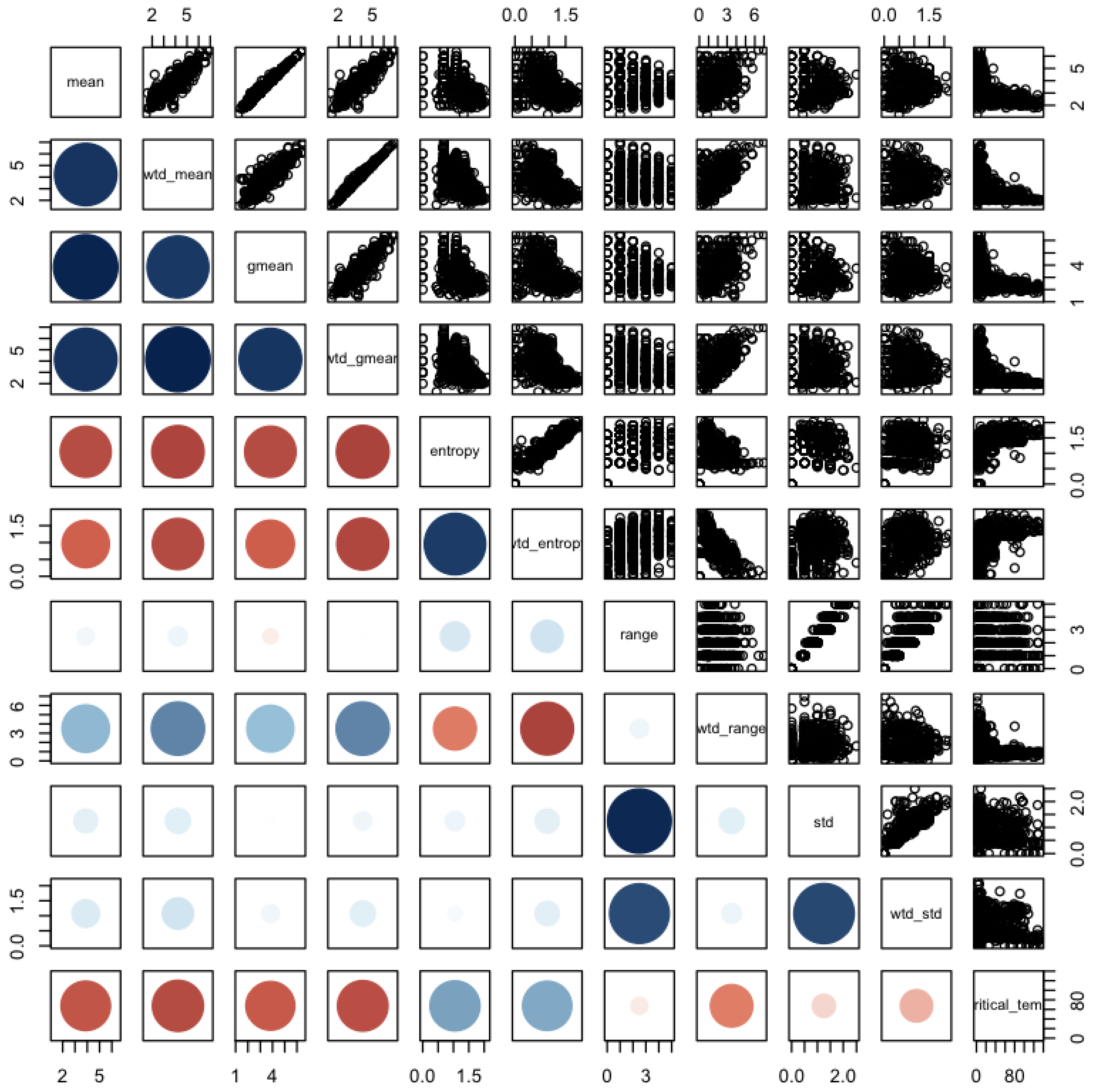
- The distribution of average values of `Thermal Conductivity` is normally distributed.
- `range`, `wtd_range` and `wtd_std` are highly positively corelated to `critical temperature`.
- `gmean` and `wtd_gmean` are highly negatively correlated with the `critical temperature`.
- `entropy` and `wtd_entropy` shows some non-colinear relationship with `critical temperature`.

Valence

```
In [27]: hist(conduct$mean_Valence)
```



```
In [28]: pairs(property8[sample.int(nrow(property8),1000),], lower.panel=panel.cor)
```



- The distribution of average values of atomic mass is normally distributed.
- `etropy` and `wtd_entropy` are positively corelated to `critical temperature`.
- `mean`, `wtd_mean`, `gmean` and `wtd_gmean` are highly negatively correlated with the `critical temperature`.
- `entropy` and `wtd_entropy` shows some non-colinear relationship with `critical temperature`.

Observations

- It seems `entropy` and `wtd_entropy` of all features shows some non-colinear relationship with `critical temperature`.
- `mean`, `wtd_mean`, `gmean` and `wtd_gmean` columns of all features are highly correlated among themselves.

3. Model Development

Since there are multiple features in our dataset, we will try different feature selection techniques to reduce the complexity of our model, filter more important properties of elements, and increase the interpretibilty/explaining power of our model.

Following methods can be used for feature selection:

- **Filter** - Features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable. We will be looking at following filter methods:
  - Pearson Correlation : Correlating the feature with the target. The features with the highest correlation are the selected.
  - MRMR(Minimum-redundancy-maximum-relevance): This feature selection method that can use either mutual information, correlation, or distance/similarity scores to select features. The aim is to penalise a feature's relevancy by its redundancy in the presence of the other selected features.
- **Wrapper** - Features are selected on the basis of information gain criteria, like AIC, BIC, AICc, etc. We will be looking at following shrinkage methods:
  - Hybrid Selection : Features are selected by combining both, 'forward' and 'backward' feature selection technique.
- **Shrinkage** - Relevant features are selected by shrinking the parameters for least important columns close to zero. We will be looking at following shrinkage methods:
  - Lasso - Absolute weight penalty term introduced in linear model.
  - Elastic Net - Absolute and squared weight penalty term introduced in linear model.

Model-1

Lasso Regression on features selected by Correlation based Feature Selection(CFS) & MRMR Selection

Variance of features

Let's compute the variance of each feature. We will remove the features with variance lesser tham **10%**, since the values from these features would be very close and wouldn't contribute highly to the our model.

```
In [29]: cols <- sort(apply(conduct, 2, var)) # evaluating variance and sorting in ascending order
         cols[1:10] # observing 10 features with least variance
         train.data.M1 <- train[cols>0.1] # filtering columns with variance higher than 10%
```

```
wtd_entropy_Electron...   0.0817881332108299
 wtd_entropy_Thermal...   0.101282043726048
    wtd_entropy_Density   0.102246922601567
 entropy_ThermalCon...    0.106260505091434
        wtd_entropy_fie   0.111567731000521
        entropy_Density   0.117207829937028
entropy_ElectronAffinity  0.117917474220004
    entropy_atomic_mass   0.133174204472861
 wtd_entropy_FusionH...   0.136992317593856
   entropy_atomic_radius  0.140933386098343
```

Observation

Only `wtd_entropy_ElectronAffinity` has variance of less than 10%. We will remove this feature from our analysis, since this isn't statistically informative for our linear model.

Correlation Based Feature Selection

In this section, we will filter out the features according to their correlation values with `critical_temp` . I have set the threshold to `0.5` , which selects even the columns with moderate correlations with `critical_temp` .

Additionally, we will ensure that the column with high variance doesn't pop up again in our analysis.

In [30]:

```
# Let's filter out features with moderate and high correlation with critical temperature
highly_correlated <- corr.m[((corr.m$Var2=='critical_temp')&(abs(corr.m$value) > 0.55)),'Var1']

print(paste('Number of features with correlation > 0.55 :',length(highly_correlated)))
print(paste("Is 'wtd_entropy_ElectronAffinity' in highly correlated features :",'wtd_entropy_ElectronAffinity' %in% highly_correlated))

# fetching names of features with high correlation to critical temperature
highly_correlated_features <- colnames(corr[,highly_correlated])
highly_correlated_features
```

```
[1] "Number of features with correlation > 0.55 : 21"
[1] "Is 'wtd_entropy_ElectronAffinity' in highly correlated features : FALSE"
```

```
'wtd_gmean_atomic_radius'  'mean_ElectronAffinity'  'wtd_gmean_atomic_mass'  'std_atomic_radius'  'range_Valence'  'mean_atomic_mass'  'wtd_mean_ElectronAffinity'  'wtd_entropy_atomic_radius'
'wtd_gmean_Density'  'wtd_gmean_Valence'  'entropy_atomic_mass'  'wtd_entropy_ElectronAffinity'  'mean_Density'  'std_Density'  'std_Valence'  'gmean_atomic_radius'  'wtd_gmean_ThermalConductivity'
'entropy_fie'  'wtd_entropy_FusionHeat'  'range_atomic_mass'  'wtd_range_ElectronAffinity'
```

Now, we will try to select the good features using **MRMR(Maximum Relevance and Minimum Redundancy)** technique. We will use `MRMR` function of `praznik` library, which inputs the data, labels and k, which is the number of features you want to select. I have used top 20 features with highest gain score.

In [31]:

```
# selcting top 20 features with highest gain score
mrmr_features <- MRMR(train.data,train.label,k=20)
data.frame(mrmr_features$score) -> mrmr_features # converting into dataframe
names(mrmr_features) <- 'score' # renaming score column
mrmr_select_feature <- row.names(mrmr_features) # extracting names of 20 features with highest gain score
mrmr_select_feature
```

```
'range_atomic_radius'  'wtd_range_ThermalConductivity'  'gmean_ElectronAffinity'  'wtd_entropy_atomic_mass'  'gmean_Valence'  'mean_ThermalConductivity'  'gmean_Density'  'wtd_range_FusionHeat'
'wtd_std_ThermalConductivity'  'std_Density'  'wtd_entropy_Valence'  'wtd_mean_Valence'  'gmean_FusionHeat'  'wtd_gmean_ElectronAffinity'  'std_ElectronAffinity'  'wtd_range_atomic_mass'
'range_ThermalConductivity'  'entropy_Density'  'wtd_gmean_Density'  'mean_Valence'
```

Now, in this analysis we will take `union` of the two set of features obtained above by two feature selection techniques, i.e. **CFS** and **MRMR**.

In [32]:

```
# union of two sets of important features
union_features <- union(highly_correlated_features,mrmr_select_feature)
print(paste('Number of features after union:',length(union_features)))

# Filter the data with relevant features
train.data.M1 <- train[,c(union_features,'critical_temp')]
head(train.data.M1)
```

```
[1] "Number of features after union: 39"
```

	wtd_gmean_atomic_radius	mean_ElectronAffinity	wtd_gmean_atomic_mass	std_atomic_radius	range_Valence	mean_atomic_mass	wtd_mean_ElectronAffinity	wtd_entropy_atomic_radius	wtd_gmean_Density	wtd_gmea
6979	85.95450	69.500	36.71000	64.61166	3	92.85524	107.48315	1.669246	62.04894	
20920	109.99415	72.240	45.77052	50.20120	3	63.32905	70.68750	1.370059	622.00587	
516	154.69254	88.570	74.63370	71.84261	1	88.56496	83.46857	1.204057	1969.09055	
2515	90.57659	56.625	35.75816	67.64963	1	86.27073	110.38831	1.585447	78.53133	
4167	89.71546	65.230	34.91444	69.42449	1	72.32465	108.69523	1.370040	66.30262	
16987	87.87166	91.350	35.55512	72.29583	2	90.14226	113.98432	1.090112	50.87849	

Let's fit a `linear` model and check the performance of the model! `lm()` function is used to fit a linear model to all the features selected above.

In [33]:
```
# fitting a linear model
fit.prelim <- lm(critical_temp~mean_Valence + entropy_Density + range_ThermalConductivity + wtd_range_atomic_mass + std_ElectronAffinity + wtd_gmean_ElectronAffinity + gmean_FusionHeat + wtd_mean_Valence + wtd_entropy_Valence + wtd_std_ThermalConductivity + wtd_range_FusionHeat + gmean_Density + mean_ThermalConductivity + gmean_Valence + wtd_entropy_atomic_mass + gmean_ElectronAffinity + wtd_range_ThermalConductivity + range_atomic_radius + wtd_range_ElectronAffinity + range_atomic_mass + wtd_entropy_FusionHeat + entropy_fie + wtd_gmean_ThermalConductivity + gmean_atomic_radius + std_Valence + std_Density + mean_Density + wtd_entropy_ElectronAffinity + entropy_atomic_mass + wtd_gmean_Valence + wtd_gmean_Density + wtd_entropy_atomic_radius + wtd_mean_ElectronAffinity + mean_atomic_mass + range_Valence + std_atomic_radius + wtd_gmean_atomic_mass + mean_ElectronAffinity + wtd_gmean_atomic_radius,
                 data=train)
summary(fit.prelim) # summary of model
```

```
Call:
lm(formula = critical_temp ~ mean_Valence + entropy_Density + 
    range_ThermalConductivity + wtd_range_atomic_mass + std_ElectronAffinity + 
    wtd_gmean_ElectronAffinity + gmean_FusionHeat + wtd_mean_Valence + 
    wtd_entropy_Valence + wtd_std_ThermalConductivity + wtd_range_FusionHeat + 
    gmean_Density + mean_ThermalConductivity + gmean_Valence + 
    wtd_entropy_atomic_mass + gmean_ElectronAffinity + wtd_range_ThermalConductivity + 
    range_atomic_radius + wtd_range_ElectronAffinity + range_atomic_mass + 
    wtd_entropy_FusionHeat + entropy_fie + wtd_gmean_ThermalConductivity + 
    gmean_atomic_radius + std_Valence + std_Density + mean_Density + 
    wtd_entropy_ElectronAffinity + entropy_atomic_mass + wtd_gmean_Valence + 
    wtd_gmean_Density + wtd_entropy_atomic_radius + wtd_mean_ElectronAffinity + 
    mean_atomic_mass + range_Valence + std_atomic_radius + wtd_gmean_atomic_mass + 
    mean_ElectronAffinity + wtd_gmean_atomic_radius, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-81.269 -10.908   0.452  11.576 112.372 

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)    
(Intercept)                    4.730e+00  3.066e+00   1.542  0.12299    
mean_Valence                   3.767e+01  4.147e+00   9.084  < 2e-16 ***
entropy_Density               -2.223e+01  2.223e+00  -9.997  < 2e-16 ***
range_ThermalConductivity     -8.974e-02  6.030e-03 -14.882  < 2e-16 ***
wtd_range_atomic_mass          4.101e-02  1.438e-02   2.852  0.00436 ** 
std_ElectronAffinity           2.407e-02  2.273e-02   1.059  0.28976    
wtd_gmean_ElectronAffinity    -2.834e-01  3.216e-02  -8.814  < 2e-16 ***
gmean_FusionHeat              -4.654e-02  3.320e-02  -1.402  0.16101    
wtd_mean_Valence              -6.758e+01  3.855e+00 -17.527  < 2e-16 ***
wtd_entropy_Valence           -6.684e+01  3.880e+00 -17.226  < 2e-16 ***
wtd_std_ThermalConductivity    2.640e-01  1.742e-02  15.155  < 2e-16 ***
wtd_range_FusionHeat           1.606e-01  2.589e-02   6.203 5.70e-10 ***
gmean_Density                 -2.105e-03  3.425e-04  -6.146 8.13e-10 ***
mean_ThermalConductivity       1.501e-01  1.267e-02  11.843  < 2e-16 ***
gmean_Valence                 -3.063e+01  3.918e+00  -7.817 5.76e-15 ***
wtd_entropy_atomic_mass        3.634e+01  2.670e+00  13.614  < 2e-16 ***
gmean_ElectronAffinity         5.564e-02  3.402e-02   1.635  0.10200    
wtd_range_ThermalConductivity  8.977e-02  1.145e-02   7.841 4.77e-15 ***
range_atomic_radius            4.758e-01  2.143e-02  22.202  < 2e-16 ***
wtd_range_ElectronAffinity    -1.698e-01  1.957e-02  -8.675  < 2e-16 ***
range_atomic_mass              4.912e-02  9.518e-03   5.161 2.49e-07 ***
wtd_entropy_FusionHeat         2.041e+01  1.615e+00  12.640  < 2e-16 ***
entropy_fie                    3.267e+01  4.018e+00   8.132 4.57e-16 ***
wtd_gmean_ThermalConductivity -1.652e-01  1.335e-02 -12.373  < 2e-16 ***
gmean_atomic_radius           -3.646e-01  2.511e-02 -14.522  < 2e-16 ***
std_Valence                   -1.546e+01  2.086e+00  -7.413 1.30e-13 ***
std_Density                   -1.465e-03  2.225e-04  -6.584 4.75e-11 ***
mean_Density                  -4.919e-04  3.137e-04  -1.568  0.11682    
wtd_entropy_ElectronAffinity  -2.534e+01  2.070e+00 -12.244  < 2e-16 ***
entropy_atomic_mass           -2.173e+01  3.852e+00  -5.643 1.70e-08 ***
wtd_gmean_Valence              6.101e+01  3.687e+00  16.548  < 2e-16 ***
wtd_gmean_Density              2.172e-03  2.558e-04   8.492  < 2e-16 ***
wtd_entropy_atomic_radius      3.710e+01  4.625e+00   8.021 1.13e-15 ***
wtd_mean_ElectronAffinity      2.493e-01  3.820e-02   6.526 6.98e-11 ***
mean_atomic_mass               9.669e-02  2.386e-02   4.052 5.10e-05 ***
range_Valence                  3.528e+00  8.155e-01   4.325 1.53e-05 ***
std_atomic_radius             -8.374e-01  5.134e-02 -16.310  < 2e-16 ***
wtd_gmean_atomic_mass         -7.288e-02  2.418e-02  -3.015  0.00258 ** 
mean_ElectronAffinity         -3.964e-02  4.320e-02  -0.918  0.35881    
wtd_gmean_atomic_radius        3.928e-01  2.236e-02  17.566  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.1 on 14844 degrees of freedom
Multiple R-squared:  0.6907,	Adjusted R-squared:  0.6899 
F-statistic:   850 on 39 and 14844 DF,  p-value: < 2.2e-16
```

Now, we will check the **Variance Inflation Factor (VIF)** of all features, which is the quotient of the variance in a model with multiple terms by the variance of a model with one term alone. It is used to filter out the features with high `multicoliinearity` in an ordinary least squares regression analysis. It provides an index that measures how much the variance of an estimated regression coefficient is increased because of collinearity. Usually, features with higher **VIF** are removed from our analysis.

In [34]:
```
# VIF of all features
sort(vif(fit.prelim))
```

```
 wtd_range_FusionHeat   3.63284778223943
     gmean_FusionHeat   4.61306175308016
          std_Density   5.66439991486909
 wtd_range_atomic_m...  6.15713890724374
 mean_ThermalCondu...   9.73206590104775
 wtd_range_ThermalC...  9.75143124399011
  std_ElectronAffinity  10.0013107593359
    range_atomic_mass   11.1044460859047
  wtd_gmean_Thermal...  11.5954793274725
   gmean_atomic_radius  12.6446613847344
 wtd_range_ElectronA... 12.8837033484195
wtd_entropy_Electron... 14.3726123179091
wtd_entropy_FusionH...  14.5834174701812
     mean_atomic_mass   20.6348262634256
      entropy_Density   23.8343903148877
 wtd_gmean_atomic_r...  26.1276213509321
         mean_Density   32.0640268481595
  wtd_gmean_atomic_...  32.2574228152754
range_ThermalCondu...   37.3100757918393
 gmean_ElectronAffinity 39.4277217120509
          std_Valence   41.4465292733434
        range_Valence   41.5602685390796
    wtd_gmean_Density   41.7803926727694
 wtd_gmean_Electron...  42.4232409101347
 wtd_entropy_atomic_... 46.9028256331877
 wtd_std_ThermalCon...  50.0274954647433
    std_atomic_radius   56.1050094011109
  mean_ElectronAffinity 58.3695319865405
 wtd_mean_ElectronA...  62.4217814870501
        gmean_Density   64.850936674275
  entropy_atomic_mass   80.7550868997325
   range_atomic_radius  84.3456334040609
  wtd_entropy_Valence   88.9716930951129
          entropy_fie   96.0501602321816
 wtd_entropy_atomic_... 144.399838621765
        gmean_Valence   676.59821600291
         mean_Valence   756.034118140417
    wtd_gmean_Valence   757.647867808532
     wtd_mean_Valence   852.903499603114
```

Some features has significantly high VIF, this indicates high collinearity of those features. Since, VIF is measure of multicollinearity, this means features with 3-digits of VIF are highly collinear with some other features. We should remove these features from our analysis.

In [35]:
```
# features with high VIF measure
high_vif <- c('wtd_entropy_atomic_radius','gmean_Valence','mean_Valence','wtd_gmean_Valence','wtd_mean_Valence')

# filtering out features that are not in above list of features with high VIF
vif_features <- union_features[!(union_features %in% high_vif)]
train.data.M1 <- train[,c(vif_features,'critical_temp')] # new filtered training dataset

fit.vif <- lm(critical_temp~.,data=train.data.M1) # fitting a model the selected features
summary(fit.vif) # summary
```

```
Call:
lm(formula = critical_temp ~ ., data = train.data.M1)

Residuals:
    Min      1Q  Median      3Q     Max 
-82.397 -11.122   0.476  11.593 115.975 

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)    
(Intercept)                   3.345e+00  2.886e+00   1.159   0.2463    
wtd_gmean_atomic_radius       2.290e-01  1.649e-02  13.886  < 2e-16 ***
mean_ElectronAffinity         9.849e-02  4.238e-02   2.324   0.0201 *  
wtd_gmean_atomic_mass        -1.680e-02  2.385e-02  -0.704   0.4812    
std_atomic_radius            -8.799e-01  5.159e-02 -17.057  < 2e-16 ***
range_Valence                 4.643e+00  8.081e-01   5.745 9.35e-09 ***
mean_atomic_mass              4.419e-02  2.341e-02   1.888   0.0591 .  
wtd_mean_ElectronAffinity     2.249e-02  3.627e-02   0.620   0.5352    
wtd_gmean_Density             1.698e-03  2.479e-04   6.847 7.83e-12 ***
entropy_atomic_mass          -2.591e+01  3.785e+00  -6.845 7.95e-12 ***
wtd_entropy_ElectronAffinity -3.691e+01  1.987e+00 -18.578  < 2e-16 ***
mean_Density                 -2.911e-04  3.017e-04  -0.965   0.3345    
std_Density                  -1.373e-03  2.230e-04  -6.156 7.65e-10 ***
std_Valence                  -1.846e+01  1.944e+00  -9.498  < 2e-16 ***
gmean_atomic_radius          -1.953e-01  2.024e-02  -9.650  < 2e-16 ***
wtd_gmean_ThermalConductivity -1.701e-01  1.314e-02 -12.947  < 2e-16 ***
entropy_fie                   3.841e+01  3.873e+00   9.917  < 2e-16 ***
wtd_entropy_FusionHeat        2.207e+01  1.554e+00  14.202  < 2e-16 ***
range_atomic_mass             5.035e-02  9.279e-03   5.427 5.83e-08 ***
wtd_range_ElectronAffinity   -1.446e-01  1.974e-02  -7.325 2.52e-13 ***
range_atomic_radius           4.787e-01  2.150e-02  22.269  < 2e-16 ***
wtd_range_ThermalConductivity 1.004e-01  1.103e-02   9.100  < 2e-16 ***
gmean_ElectronAffinity        3.998e-03  3.416e-02   0.117   0.9069    
wtd_entropy_atomic_mass       4.845e+01  2.094e+00  23.135  < 2e-16 ***
mean_ThermalConductivity      1.440e-01  1.258e-02  11.443  < 2e-16 ***
gmean_Density                -1.786e-03  3.416e-04  -5.228 1.73e-07 ***
wtd_range_FusionHeat          1.870e-01  2.557e-02   7.312 2.76e-13 ***
wtd_std_ThermalConductivity   2.644e-01  1.714e-02  15.426  < 2e-16 ***
wtd_entropy_Valence          -3.264e+01  2.396e+00 -13.624  < 2e-16 ***
gmean_FusionHeat              1.770e-02  3.268e-02   0.542   0.5881    
wtd_gmean_ElectronAffinity   -1.394e-01  3.127e-02  -4.459 8.29e-06 ***
std_ElectronAffinity         -5.043e-03  2.265e-02  -0.223   0.8239    
wtd_range_atomic_mass         4.562e-02  1.446e-02   3.156   0.0016 ** 
range_ThermalConductivity    -8.354e-02  5.998e-03 -13.929  < 2e-16 ***
entropy_Density              -2.500e+01  2.191e+00 -11.409  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.32 on 14849 degrees of freedom
Multiple R-squared:  0.6834,	Adjusted R-squared:  0.6827 
F-statistic: 942.7 on 34 and 14849 DF,  p-value: < 2.2e-16
```

It is evident from the R-squared values of the `fit.prelim` , which is 0.6899, and `fit.vif` (linear model after removing features with high VIF), which is 0.6827, that those features were multicollinear. R-squared value changed in third decimal place, which indicates the fit of our linear model isn't affected much after removing those features. Upon further inspection of the model, we can see that there are few features that have high `p-value` . Those feature selected by techniques used above, are not statistically significant. `p-value` in the stats above, indicates the probability of the test statistic at least as unusual as the one we obtained, if the null hypothesis were true. In this case, the null hypothesis is that the true coefficient is zero; if that probability is low, it's suggesting that it would be rare to get a result as unusual as this if the coefficient were really zero. However, the features with comparatively higher p-value, aren't statistically useful and can be removed from our analysis.

To remove the features with lower statistical significance (possibly features with p-values close to 0), we will use **`Wrapper`** method of feature selection. Wrapper methods are based on greedy search algorithms, as they evaluate all possible combinations of the features and select the combination that produces the best result for a specific machine learning algorithm, using some information gain criteria. By default it uses `Akaike Information Criteria (AIC)` , and continues iteration until the AIC stops reducing with the change of features. This can be achived by using `step()` function in R. We will use `Hybrid feature selection` algorithm by setting the argument `direction = 'both'` in step() funcition.

In [36]:

```
# Hybrid feature selection
fit.step <- step(fit.vif,direction='both')
summary(fit.step)
```

```
Start:  AIC=88185.89
critical_temp ~ wtd_gmean_atomic_radius + mean_ElectronAffinity + 
    wtd_gmean_atomic_mass + std_atomic_radius + range_Valence + 
    mean_atomic_mass + wtd_mean_ElectronAffinity + wtd_gmean_Density + 
    entropy_atomic_mass + wtd_entropy_ElectronAffinity + mean_Density + 
    std_Density + std_Valence + gmean_atomic_radius + wtd_gmean_ThermalConductivity + 
    entropy_fie + wtd_entropy_FusionHeat + range_atomic_mass + 
    wtd_range_ElectronAffinity + range_atomic_radius + wtd_range_ThermalConductivity + 
    gmean_ElectronAffinity + wtd_entropy_atomic_mass + mean_ThermalConductivity + 
    gmean_Density + wtd_range_FusionHeat + wtd_std_ThermalConductivity + 
    wtd_entropy_Valence + gmean_FusionHeat + wtd_gmean_ElectronAffinity + 
    std_ElectronAffinity + wtd_range_atomic_mass + range_ThermalConductivity + 
    entropy_Density

                                Df Sum of Sq     RSS   AIC
- gmean_ElectronAffinity         1         5 5543952 88184
- std_ElectronAffinity           1        18 5543965 88184
- gmean_FusionHeat               1       110 5544056 88184
- wtd_mean_ElectronAffinity      1       144 5544090 88184
- wtd_gmean_atomic_mass          1       185 5544132 88184
- mean_Density                   1       348 5544295 88185
<none>                                         5543947 88186
- mean_atomic_mass               1      1331 5545278 88187
- mean_ElectronAffinity          1      2016 5545963 88189
- wtd_range_atomic_mass          1      3719 5547665 88194
- wtd_gmean_ElectronAffinity     1      7424 5551370 88204
- gmean_Density                  1     10206 5554153 88211
- range_atomic_mass              1     10995 5554941 88213
- range_Valence                  1     12324 5556271 88217
- std_Density                    1     14149 5558096 88222
- entropy_atomic_mass            1     17493 5561440 88231
- wtd_gmean_Density              1     17504 5561451 88231
- wtd_range_FusionHeat           1     19963 5563910 88237
- wtd_range_ElectronAffinity     1     20031 5563977 88238
- wtd_range_ThermalConductivity  1     30915 5574862 88267
- std_Valence                    1     33679 5577626 88274
- gmean_atomic_radius            1     34767 5578714 88277
- entropy_fie                    1     36720 5580666 88282
- entropy_Density                1     48602 5592549 88314
- mean_ThermalConductivity       1     48884 5592831 88315
- wtd_gmean_ThermalConductivity  1     62588 5606535 88351
- wtd_entropy_Valence            1     69305 5613251 88369
- wtd_gmean_atomic_radius        1     71991 5615937 88376
- range_ThermalConductivity      1     72441 5616388 88377
- wtd_entropy_FusionHeat         1     75302 5619249 88385
- wtd_std_ThermalConductivity    1     88847 5632794 88421
- std_atomic_radius              1    108625 5652571 88473
- wtd_entropy_ElectronAffinity   1    128864 5672811 88526
- range_atomic_radius            1    185146 5729093 88673
- wtd_entropy_atomic_mass        1    199834 5743781 88711

Step:  AIC=88183.9
critical_temp ~ wtd_gmean_atomic_radius + mean_ElectronAffinity + 
    wtd_gmean_atomic_mass + std_atomic_radius + range_Valence + 
    mean_atomic_mass + wtd_mean_ElectronAffinity + wtd_gmean_Density + 
    entropy_atomic_mass + wtd_entropy_ElectronAffinity + mean_Density + 
    std_Density + std_Valence + gmean_atomic_radius + wtd_gmean_ThermalConductivity + 
    entropy_fie + wtd_entropy_FusionHeat + range_atomic_mass + 
    wtd_range_ElectronAffinity + range_atomic_radius + wtd_range_ThermalConductivity + 
    wtd_entropy_atomic_mass + mean_ThermalConductivity + gmean_Density + 
    wtd_range_FusionHeat + wtd_std_ThermalConductivity + wtd_entropy_Valence + 
    gmean_FusionHeat + wtd_gmean_ElectronAffinity + std_ElectronAffinity + 
    wtd_range_atomic_mass + range_ThermalConductivity + entropy_Density

                                Df Sum of Sq     RSS   AIC
- std_ElectronAffinity           1        65 5544017 88182
- gmean_FusionHeat               1       108 5544060 88182
- wtd_mean_ElectronAffinity      1       150 5544102 88182
- wtd_gmean_atomic_mass          1       191 5544143 88182
- mean_Density                   1       356 5544308 88183
<none>                                         5543952 88184
- mean_atomic_mass               1      1339 5545291 88185
+ gmean_ElectronAffinity         1         5 5543947 88186
- wtd_range_atomic_mass          1      3783 5547735 88192
- wtd_gmean_ElectronAffinity     1      9746 5553698 88208
- gmean_Density                  1     10216 5554168 88209
- mean_ElectronAffinity          1     10890 5554842 88211
- range_atomic_mass              1     11028 5554980 88211
- range_Valence                  1     12423 5556375 88215
- std_Density                    1     14146 5558098 88220
- wtd_gmean_Density              1     17538 5561490 88229
- entropy_atomic_mass            1     17845 5561798 88230
- wtd_range_FusionHeat           1     19969 5563921 88235
- wtd_range_ElectronAffinity     1     20076 5564028 88236
- wtd_range_ThermalConductivity  1     30928 5574880 88265
- std_Valence                    1     33804 5577756 88272
- gmean_atomic_radius            1     34795 5578747 88275
- entropy_fie                    1     36810 5580762 88280
- entropy_Density                1     48622 5592574 88312
- mean_ThermalConductivity       1     49293 5593245 88314
- wtd_gmean_ThermalConductivity  1     62591 5606543 88349
- wtd_entropy_Valence            1     69744 5613696 88368
- range_ThermalConductivity      1     72613 5616565 88376
- wtd_gmean_atomic_radius        1     73277 5617229 88377
- wtd_entropy_FusionHeat         1     75316 5619268 88383
- wtd_std_ThermalConductivity    1     89175 5633127 88419
- std_atomic_radius              1    108706 5652658 88471
- wtd_entropy_ElectronAffinity   1    129726 5673678 88526
- range_atomic_radius            1    185172 5729124 88671
- wtd_entropy_atomic_mass        1    206293 5750245 88726

Step:  AIC=88182.08
critical_temp ~ wtd_gmean_atomic_radius + mean_ElectronAffinity + 
    wtd_gmean_atomic_mass + std_atomic_radius + range_Valence + 
    mean_atomic_mass + wtd_mean_ElectronAffinity + wtd_gmean_Density + 
    entropy_atomic_mass + wtd_entropy_ElectronAffinity + mean_Density + 
    std_Density + std_Valence + gmean_atomic_radius + wtd_gmean_ThermalConductivity + 
    entropy_fie + wtd_entropy_FusionHeat + range_atomic_mass + 
    wtd_range_ElectronAffinity + range_atomic_radius + wtd_range_ThermalConductivity + 
    wtd_entropy_atomic_mass + mean_ThermalConductivity + gmean_Density + 
    wtd_range_FusionHeat + wtd_std_ThermalConductivity + wtd_entropy_Valence + 
    gmean_FusionHeat + wtd_gmean_ElectronAffinity + wtd_range_atomic_mass + 
    range_ThermalConductivity + entropy_Density

                                Df Sum of Sq     RSS   AIC
- wtd_mean_ElectronAffinity      1       107 5544124 88180
- gmean_FusionHeat               1       107 5544125 88180
- wtd_gmean_atomic_mass          1       217 5544234 88181
- mean_Density                   1       348 5544366 88181
<none>                                         5544017 88182
- mean_atomic_mass               1      1407 5545424 88184
+ std_ElectronAffinity           1        65 5543952 88184
+ gmean_ElectronAffinity         1        52 5543965 88184
- wtd_range_atomic_mass          1      3902 5547919 88191
- gmean_Density                  1     10351 5554368 88208
- range_atomic_mass              1     10972 5554989 88210
- mean_ElectronAffinity          1     12245 5556262 88213
- range_Valence                1     12359 5556376 88213
- wtd_gmean_ElectronAffinity     1     14169 5558186 88218
- std_Density                    1     14472 5558490 88219
- wtd_gmean_Density              1     17597 5561614 88227
- entropy_atomic_mass            1     18047 5562065 88228
- wtd_range_FusionHeat           1     20076 5564094 88234
- wtd_range_ElectronAffinity     1     23931 5567948 88244
- wtd_range_ThermalConductivity  1     31046 5575063 88263
- std_Valence                    1     33816 5577833 88271
- gmean_atomic_radius            1     34775 5578792 88273
- entropy_fie                    1     36821 5580838 88279
- entropy_Density                1     48563 5592581 88310
- mean_ThermalConductivity       1     50209 5594226 88314
- wtd_gmean_ThermalConductivity  1     63222 5607239 88349
- wtd_entropy_Valence            1     70917 5614934 88369
- range_ThermalConductivity      1     73033 5617050 88375
- wtd_entropy_FusionHeat         1     76262 5620279 88383
- wtd_gmean_atomic_radius        1     76282 5620299 88383
- wtd_std_ThermalConductivity    1     90262 5634279 88420
- std_atomic_radius              1    108654 5652671 88469
- wtd_entropy_ElectronAffinity   1    142699 5686717 88558
- range_atomic_radius            1    185192 5729210 88669
- wtd_entropy_atomic_mass        1    206548 5750566 88725

Step:  AIC=88180.36
critical_temp ~ wtd_gmean_atomic_radius + mean_ElectronAffinity + 
    wtd_gmean_atomic_mass + std_atomic_radius + range_Valence + 
    mean_atomic_mass + wtd_gmean_Density + entropy_atomic_mass + 
    wtd_entropy_ElectronAffinity + mean_Density + std_Density + 
    std_Valence + gmean_atomic_radius + wtd_gmean_ThermalConductivity + 
    entropy_fie + wtd_entropy_FusionHeat + range_atomic_mass + 
    wtd_range_ElectronAffinity + range_atomic_radius + wtd_range_ThermalConductivity + 
    wtd_entropy_atomic_mass + mean_ThermalConductivity + gmean_Density + 
    wtd_range_FusionHeat + wtd_std_ThermalConductivity + wtd_entropy_Valence + 
    gmean_FusionHeat + wtd_gmean_ElectronAffinity + wtd_range_atomic_mass + 
    range_ThermalConductivity + entropy_Density

                                Df Sum of Sq     RSS   AIC
- gmean_FusionHeat               1       127 5544251 88179
- wtd_gmean_atomic_mass          1       179 5544304 88179
- mean_Density                   1       398 5544522 88179
<none>                                       5544124 88180
- mean_atomic_mass               1      1356 5545481 88182
+ wtd_mean_ElectronAffinity      1       107 5544017 88182
+ std_ElectronAffinity           1        23 5544102 88182
+ gmean_ElectronAffinity         1         1 5544123 88182
- wtd_range_atomic_mass          1      3795 5547920 88189
- gmean_Density                  1     10310 5554434 88206
- range_atomic_mass              1     11504 5555628 88209
- range_Valence                  1     12284 5556408 88211
- std_Density                    1     14566 5558691 88217
- wtd_gmean_Density              1     17788 5561912 88226
- entropy_atomic_mass            1     18084 5562209 88227
- wtd_range_FusionHeat           1     20104 5564229 88232
- mean_ElectronAffinity          1     25148 5569273 88246
- wtd_range_ElectronAffinity     1     30602 5574726 88260
- wtd_range_ThermalConductivity  1     31049 5575174 88261
- wtd_gmean_ElectronAffinity     1     31600 5575724 88263
- std_Valence                    1     33709 5577833 88269
- gmean_atomic_radius            1     34685 5578809 88271
- entropy_fie                    1     37455 5581580 88279
- mean_ThermalConductivity       1     50261 5594386 88313
- entropy_Density                1     51357 5595481 88316
- wtd_gmean_ThermalConductivity  1     63119 5607243 88347
- wtd_entropy_Valence            1     74459 5618583 88377
- range_ThermalConductivity      1     74823 5618947 88378
- wtd_gmean_atomic_radius        1     76622 5620746 88383
- wtd_entropy_FusionHeat         1     77052 5621177 88384
- wtd_std_ThermalConductivity    1     93943 5638067 88428
- std_atomic_radius              1    114004 5658129 88481
- wtd_entropy_ElectronAffinity   1    160985 5705109 88604
- range_atomic_radius            1    191624 5735749 88684
- wtd_entropy_atomic_mass        1    206657 5750782 88723

Step:  AIC=88178.7
critical_temp ~ wtd_gmean_atomic_radius + mean_ElectronAffinity + 
    wtd_gmean_atomic_mass + std_atomic_radius + range_Valence + 
    mean_atomic_mass + wtd_gmean_Density + entropy_atomic_mass + 
    wtd_entropy_ElectronAffinity + mean_Density + std_Density + 
    std_Valence + gmean_atomic_radius + wtd_gmean_ThermalConductivity + 
    entropy_fie + wtd_entropy_FusionHeat + range_atomic_mass + 
    wtd_range_ElectronAffinity + range_atomic_radius + wtd_range_ThermalConductivity + 
    wtd_entropy_atomic_mass + mean_ThermalConductivity + gmean_Density + 
    wtd_range_FusionHeat + wtd_std_ThermalConductivity + wtd_entropy_Valence + 
    wtd_gmean_ElectronAffinity + wtd_range_atomic_mass + range_ThermalConductivity + 
    entropy_Density

                                Df Sum of Sq     RSS   AIC
- wtd_gmean_atomic_mass          1       287 5544538 88177
- mean_Density                   1       444 5544696 88178
<none>                                       5544251 88179
+ gmean_FusionHeat               1       127 5544124 88180
+ wtd_mean_ElectronAffinity      1       127 5544125 88180
- mean_atomic_mass               1      1366 5545618 88180
+ std_ElectronAffinity           1        20 5544232 88181
+ gmean_ElectronAffinity         1         0 5544251 88181
- wtd_range_atomic_mass          1      4054 5548305 88188
- gmean_Density                  1     10241 5554492 88204
- range_atomic_mass              1     11385 5555636 88207
- range_Valence                  1     12502 5556754 88210
- std_Density                    1     14499 5558750 88216
- entropy_atomic_mass            1     18207 5562459 88226
- wtd_gmean_Density              1     18301 5562552 88226
- wtd_range_FusionHeat           1     23899 5568150 88241
- mean_ElectronAffinity          1     25520 5569772 88245
- wtd_range_ThermalConductivity  1     31428 5575679 88261
- wtd_gmean_ElectronAffinity     1     32033 5576285 88262
- std_Valence                    1     34250 5578502 88268
- wtd_range_ElectronAffinity     1     34660 5578911 88269
- gmean_atomic_radius            1     35524 5579775 88272
- entropy_fie                    1     37362 5581614 88277
- mean_ThermalConductivity       1     50135 5594386 88311
- entropy_Density                1     51705 5595957 88315
- wtd_gmean_ThermalConductivity  1     63739 5607991 88347
- wtd_entropy_Valence            1     75046 5619298 88377
- range_ThermalConductivity      1     76489 5620741 88381
- wtd_gmean_atomic_radius        1     78048 5622299 88385
- wtd_entropy_FusionHeat         1     86703 5630954 88408
- wtd_std_ThermalConductivity    1     97532 5641783 88436
- std_atomic_radius              1    114416 5658667 88481
- wtd_entropy_ElectronAffinity   1    175197 5719448 88640
- range_atomic_radius            1    193711 5737963 88688
- wtd_entropy_atomic_mass        1    209108 5753359 88728

Step:  AIC=88177.47
critical_temp ~ wtd_gmean_atomic_radius + mean_ElectronAffinity + 
    std_atomic_radius + range_Valence + mean_atomic_mass + wtd_gmean_Density + 
    entropy_atomic_mass + wtd_entropy_ElectronAffinity + mean_Density +
    std_Density + std_Valence + gmean_atomic_radius + wtd_gmean_ThermalConductivity +
    entropy_fie + wtd_entropy_FusionHeat + range_atomic_mass +
    wtd_range_ElectronAffinity + range_atomic_radius + wtd_range_ThermalConductivity +
    wtd_entropy_atomic_mass + mean_ThermalConductivity + gmean_Density +
    wtd_range_FusionHeat + wtd_std_ThermalConductivity + wtd_entropy_Valence +
    wtd_gmean_ElectronAffinity + wtd_range_atomic_mass + range_ThermalConductivity +
    entropy_Density

                                Df Sum of Sq     RSS   AIC
- mean_Density                   1       288 5544826 88176
<none>                                       5544538 88177
+ wtd_gmean_atomic_mass          1       287 5544251 88179
+ gmean_FusionHeat               1       234 5544304 88179
+ wtd_mean_ElectronAffinity      1        82 5544456 88179
+ std_ElectronAffinity           1        46 5544492 88179
+ gmean_ElectronAffinity         1        10 5544529 88179
- mean_atomic_mass               1      1578 5546116 88180
- wtd_range_atomic_mass          1      3771 5548309 88186
- gmean_Density                  1      9996 5554535 88202
- range_Valence                  1     12219 5556757 88208
- range_atomic_mass              1     13755 5558293 88212
- std_Density                    1     14640 5559178 88215
- entropy_atomic_mass            1     20743 5565281 88231
- wtd_gmean_Density              1     22794 5567333 88237
- wtd_range_FusionHeat           1     23810 5568348 88239
- mean_ElectronAffinity          1     26513 5571051 88246
- wtd_range_ThermalConductivity  1     31236 5575774 88259
- wtd_gmean_ElectronAffinity     1     33170 5577708 88264
- std_Valence                    1     34071 5578609 88267
- wtd_range_ElectronAffinity     1     34488 5579026 88268
- gmean_atomic_radius            1     38337 5582875 88278
- entropy_fie                    1     40269 5584807 88283
- mean_ThermalConductivity       1     50963 5595501 88312
- entropy_Density                1     52252 5596790 88315
- wtd_gmean_ThermalConductivity  1     66429 5610967 88353
- wtd_entropy_Valence            1     76166 5620704 88379
- range_ThermalConductivity      1     76245 5620783 88379
- wtd_entropy_FusionHeat         1     87257 5631795 88408
- wtd_gmean_atomic_radius        1     91146 5635684 88418
- wtd_std_ThermalConductivity    1     98501 5643039 88438
- std_atomic_radius              1    115860 5660398 88483
- wtd_entropy_ElectronAffinity   1    175239 5719777 88639
- range_atomic_radius            1    195806 5740344 88692
- wtd_entropy_atomic_mass        1    210151 5754689 88729

Step:  AIC=88176.25
critical_temp ~ wtd_gmean_atomic_radius + mean_ElectronAffinity +
    std_atomic_radius + range_Valence + mean_atomic_mass + wtd_gmean_Density +
    entropy_atomic_mass + wtd_entropy_ElectronAffinity + std_Density +
    std_Valence + gmean_atomic_radius + wtd_gmean_ThermalConductivity +
    entropy_fie + wtd_entropy_FusionHeat + range_atomic_mass +
    wtd_range_ElectronAffinity + range_atomic_radius + wtd_range_ThermalConductivity +
    wtd_entropy_atomic_mass + mean_ThermalConductivity + gmean_Density +
    wtd_range_FusionHeat + wtd_std_ThermalConductivity + wtd_entropy_Valence +
    wtd_gmean_ElectronAffinity + wtd_range_atomic_mass + range_ThermalConductivity +
    entropy_Density

                                Df Sum of Sq     RSS   AIC
<none>                                       5544826 88176
+ mean_Density                   1       288 5544538 88177
+ gmean_FusionHeat               1       243 5544583 88178
- mean_atomic_mass               1      1319 5546145 88178
+ wtd_mean_ElectronAffinity      1       136 5544690 88178
+ wtd_gmean_atomic_mass          1       130 5544696 88178
+ std_ElectronAffinity           1        25 5544801 88178
+ gmean_ElectronAffinity         1         3 5544823 88178
- wtd_range_atomic_mass          1      3702 5548527 88184
- range_Valence                  1     12044 5556870 88207
- range_atomic_mass              1     13766 5558592 88211
- entropy_atomic_mass            1     21449 5566275 88232
- wtd_gmean_Density              1     23240 5568066 88236
- wtd_range_FusionHeat           1     24240 5569066 88239
- gmean_Density                  1     25244 5570070 88242
- mean_ElectronAffinity          1     27741 5572567 88249
- std_Density                    1     29043 5573869 88252
- wtd_range_ThermalConductivity  1     31336 5576161 88258
- wtd_gmean_ElectronAffinity     1     33421 5578246 88264
- std_Valence                    1     33855 5578681 88265
- wtd_range_ElectronAffinity     1     34364 5579190 88266
- gmean_atomic_radius            1     38432 5583258 88277
- entropy_fie                    1     42724 5587549 88288
- mean_ThermalConductivity       1     51951 5596777 88313
- entropy_Density                1     52144 5596970 88314
- wtd_gmean_ThermalConductivity  1     66659 5611485 88352
- wtd_entropy_Valence            1     77349 5622175 88380
- range_ThermalConductivity      1     81299 5626125 88391
- wtd_entropy_FusionHeat         1     87647 5632473 88408
- wtd_gmean_atomic_radius        1     91424 5636250 88418
- wtd_std_ThermalConductivity    1     99187 5644013 88438
- std_atomic_radius              1    119702 5664528 88492
- wtd_entropy_ElectronAffinity   1    176412 5721238 88640
- range_atomic_radius            1    209916 5754742 88727
- wtd_entropy_atomic_mass        1    213142 5757968 88736
```

```
Call:
lm(formula = critical_temp ~ wtd_gmean_atomic_radius + mean_ElectronAffinity +
    std_atomic_radius + range_Valence + mean_atomic_mass + wtd_gmean_Density +
    entropy_atomic_mass + wtd_entropy_ElectronAffinity + std_Density +
    std_Valence + gmean_atomic_radius + wtd_gmean_ThermalConductivity +
    entropy_fie + wtd_entropy_FusionHeat + range_atomic_mass +
    wtd_range_ElectronAffinity + range_atomic_radius + wtd_range_ThermalConductivity +
    wtd_entropy_atomic_mass + mean_ThermalConductivity + gmean_Density +
    wtd_range_FusionHeat + wtd_std_ThermalConductivity + wtd_entropy_Valence +
    wtd_gmean_ElectronAffinity + wtd_range_atomic_mass + range_ThermalConductivity +
    entropy_Density, data = train.data.M1)

Residuals:
    Min      1Q  Median      3Q     Max
-82.171 -11.114   0.459  11.578 115.660

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    2.476e+00  2.637e+00   0.939  0.34784
wtd_gmean_atomic_radius        2.246e-01  1.435e-02  15.650  < 2e-16 ***
mean_ElectronAffinity          1.093e-01  1.267e-02   8.621  < 2e-16 ***
std_atomic_radius             -8.679e-01  4.846e-02 -17.908  < 2e-16 ***
range_Valence                  4.484e+00  7.893e-01   5.680 1.37e-08 ***
mean_atomic_mass               2.437e-02  1.296e-02   1.880  0.06013 .
wtd_gmean_Density              1.616e-03  2.048e-04   7.891 3.22e-15 ***
entropy_atomic_mass           -2.731e+01  3.603e+00  -7.581 3.64e-14 ***
wtd_entropy_ElectronAffinity  -3.783e+01  1.740e+00 -21.740  < 2e-16 ***
std_Density                   -1.492e-03  1.692e-04  -8.821  < 2e-16 ***
std_Valence                   -1.810e+01  1.901e+00  -9.524  < 2e-16 ***
gmean_atomic_radius           -1.863e-01  1.836e-02 -10.147  < 2e-16 ***
wtd_gmean_ThermalConductivity -1.662e-01  1.244e-02 -13.364  < 2e-16 ***
entropy_fie                    3.965e+01  3.706e+00  10.699  < 2e-16 ***
wtd_entropy_FusionHeat         2.229e+01  1.455e+00  15.324  < 2e-16 ***
range_atomic_mass              5.278e-02  8.692e-03   6.073 1.29e-09 ***
wtd_range_ElectronAffinity    -1.442e-01  1.503e-02  -9.595  < 2e-16 ***
range_atomic_radius            4.731e-01  1.995e-02  23.715  < 2e-16 ***
wtd_range_ThermalConductivity  9.732e-02  1.062e-02   9.162  < 2e-16 ***
wtd_entropy_atomic_mass        4.860e+01  2.034e+00  23.896  < 2e-16 ***
mean_ThermalConductivity       1.426e-01  1.208e-02  11.797  < 2e-16 ***
gmean_Density                 -1.928e-03  2.345e-04  -8.224  < 2e-16 ***
wtd_range_FusionHeat           1.934e-01  2.399e-02   8.059 8.30e-16 ***
wtd_std_ThermalConductivity    2.681e-01  1.644e-02  16.301  < 2e-16 ***
wtd_entropy_Valence           -3.225e+01  2.240e+00 -14.395  < 2e-16 ***
wtd_gmean_ElectronAffinity    -1.220e-01  1.289e-02  -9.462  < 2e-16 ***
wtd_range_atomic_mass          4.217e-02  1.339e-02   3.149  0.00164 **
range_ThermalConductivity     -8.465e-02  5.736e-03 -14.758  < 2e-16 ***
entropy_Density               -2.462e+01  2.083e+00 -11.819  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.32 on 14855 degrees of freedom
Multiple R-squared:  0.6833,    Adjusted R-squared:  0.6827
F-statistic:  1145 on 28 and 14855 DF,  p-value: < 2.2e-16
```

```
In [37]: # saving the names of features of model 1
         model_1_features <- names(fit.step$coefficients)[-1]
```

Looking at the model above, we can see that all the staistical non-significant columns are removed, without changing R-squared value at all. `28` features are selected in the final model. Now, we will check the performance of this model on our test set as follows.

```
In [38]: # making predictions using fitted model above
         test_pred <- predict(fit.step,newdata=test.data)
         # Checking MSE
         mse_model_1 <- sqrt(mean((test.label - test_pred)^2))
         print(paste('RMSE for Model 1:',mse_model_1))

         # Checking R-Squared Value
         rsq_model_1 <- cor(test.label, test_pred)^2
         print(paste('R-Squared for Model 1:',rsq_model_1))
```

```
[1] "RMSE for Model 1: 19.3971764200639"
[1] "R-Squared for Model 1: 0.677489175656799"
```

```
In [39]: # data frame to store performance of model on test set
         model_comp <- data.frame('Model' = rep(0,4),
                                  'R-Squared' = rep(0,4),
                                  'R.M.S.E' = rep(0,4),
                                  'Features' = rep(0,4))
         model_comp[1,] <- c('Linear model + CFS + MRMR', round(rsq_model_1,3),round(mse_model_1,3),length(fit.step$coefficients)-1)
```

Since, we can't select features by visual inspection or manually,we will use `Regularization` methods for our rescue.

Model - 2

Linear model with Regularization

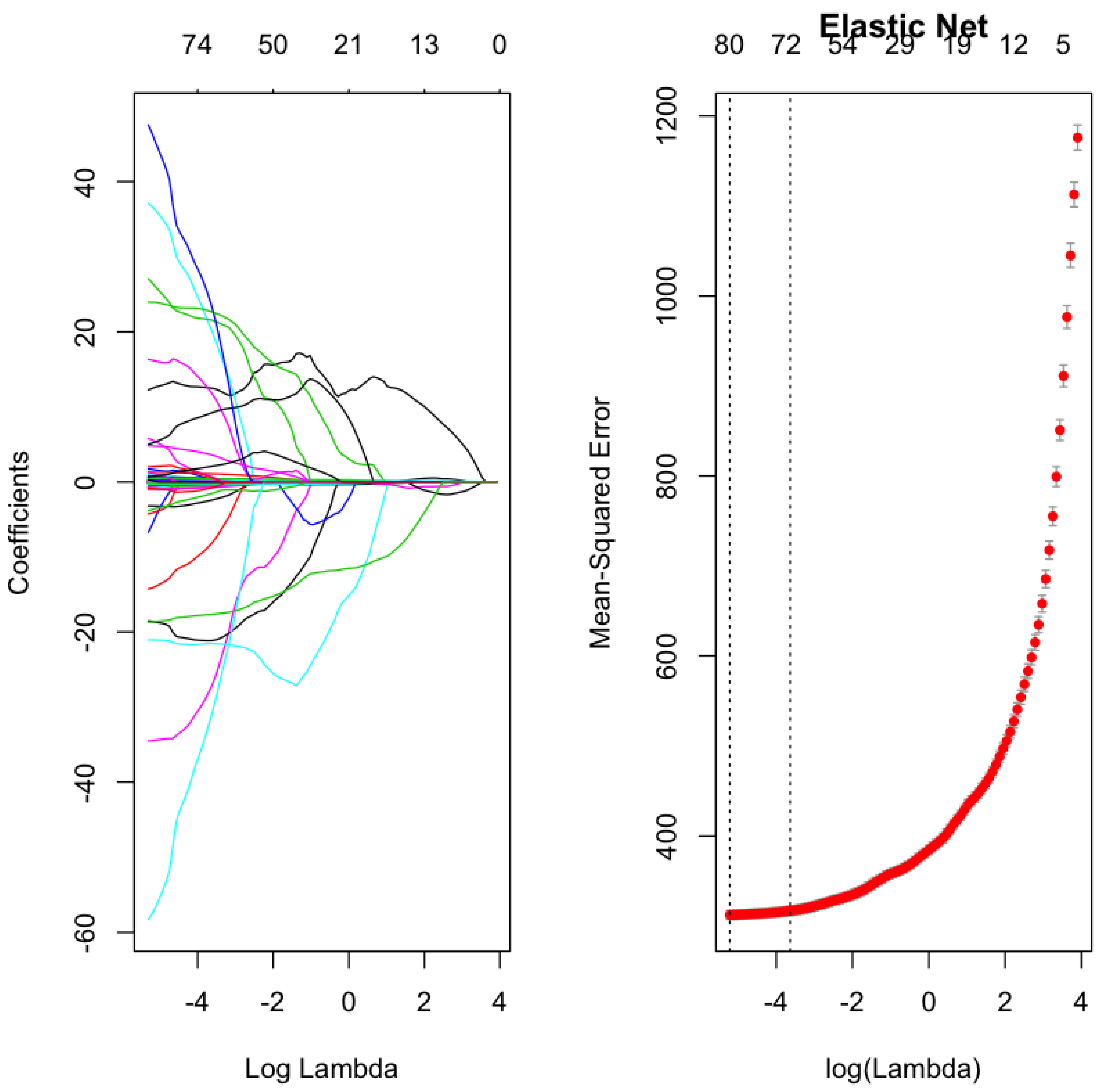
In this section we will use `shrinkage` methods, to choose the most relevant features from our dataset, by significantly shrinking the parameters of non-relevant features, and in-turn shrinking their effect in our model.

Elastic net

First, we will check the performance of `Elastic Net` . Elastic Net combines the power of two popular regularization techniques, i.e. `Lasso` and `Ridge` . It is most useful when there are many features in the dataset, and you want to filter out the best one by significantly shrinking the parameters of non-relevant features. This method can be implemented by `glmnet` function of glmnet library, and using `alpha = 0.5` as a parameter.

```
In [40]: #  fitting elastic net regression to training data set
         fit.elnet <- glmnet(as.matrix(train.data), as.matrix(train.label), family="gaussian", alpha=.5)
         fit.elnet.cv <- cv.glmnet(as.matrix(train.data), as.matrix(train.label), type.measure="mse", alpha=.5,
                                   family="gaussian")
```

In [41]:
```
# Plot solution paths:
par(mfrow=c(1,2))
plot(fit.elnet,xvar="lambda")
plot(fit.elnet.cv, main="Elastic Net")
```



In [42]:
```
# making predictions using fitted model above
test_pred <- predict(fit.elnet.cv, s=fit.elnet.cv$lambda.1se, newx=as.matrix(test.data))
# Checking MSE
mse_elnet <- sqrt(mean((test.label - test_pred)^2))
print(paste('RMSE for Elastic Net:',mse_elnet))

# Checking R-Squared Value
rsq_elnet <- cor(test.label, test_pred)^2
print(paste('R-Squared for Elastic Net:',rsq_elnet))
```

```
[1] "RMSE for Elastic Net: 17.8867776130901"
[1] "R-Squared for Elastic Net: 0.725616277750196"
```

In [43]:
```
data.frame(predict(fit.elnet.cv, s = fit.elnet.cv$lambda.1se, type = "coefficients")[1:82,]) -> features.elnet
names(features.elnet) <- 'weight'
Features <- row.names(features.elnet)
row.names(features.elnet) <- 1:nrow(features.elnet)
features.elnet<-cbind(Features,features.elnet)
# Check the features with zero weights
features.elnet[(abs(features.elnet$weight)< 0.05),]

#storing values
model_comp[2,] <- c('Elastic Net', round(rsq_elnet,3),round(mse_elnet,3),nrow(features.elnet[(abs(features.elnet$weight)!=0),])-1)
```

	Features	weight
5	gmean_atomic_mass	-4.972665e-03
6	wtd_gmean_atomic_mass	-6.368910e-03
10	wtd_range_atomic_mass	0.000000e+00
13	mean_fie	5.294658e-03
14	wtd_mean_fie	1.532681e-02
15	gmean_fie	0.000000e+00
16	wtd_gmean_fie	0.000000e+00
17	entropy_fie	0.000000e+00
20	wtd_range_fie	1.413614e-02
22	wtd_std_fie	-2.553024e-02
27	entropy_atomic_radius	0.000000e+00
33	mean_Density	-2.576916e-03
34	wtd_mean_Density	-3.025306e-05
35	gmean_Density	0.000000e+00
36	wtd_gmean_Density	1.453054e-03
39	range_Density	-9.309988e-04
40	wtd_range_Density	9.008263e-05
41	std_Density	3.285564e-03
42	wtd_std_Density	-1.394971e-03
43	mean_ElectronAffinity	-2.591829e-02
55	gmean_FusionHeat	-4.102584e-02
56	wtd_gmean_FusionHeat	0.000000e+00
63	mean_ThermalConductivity	0.000000e+00
69	range_ThermalConductivity	-3.347428e-02
71	std_ThermalConductivity	3.354627e-02
73	mean_Valence	0.000000e+00
74	wtd_mean_Valence	0.000000e+00

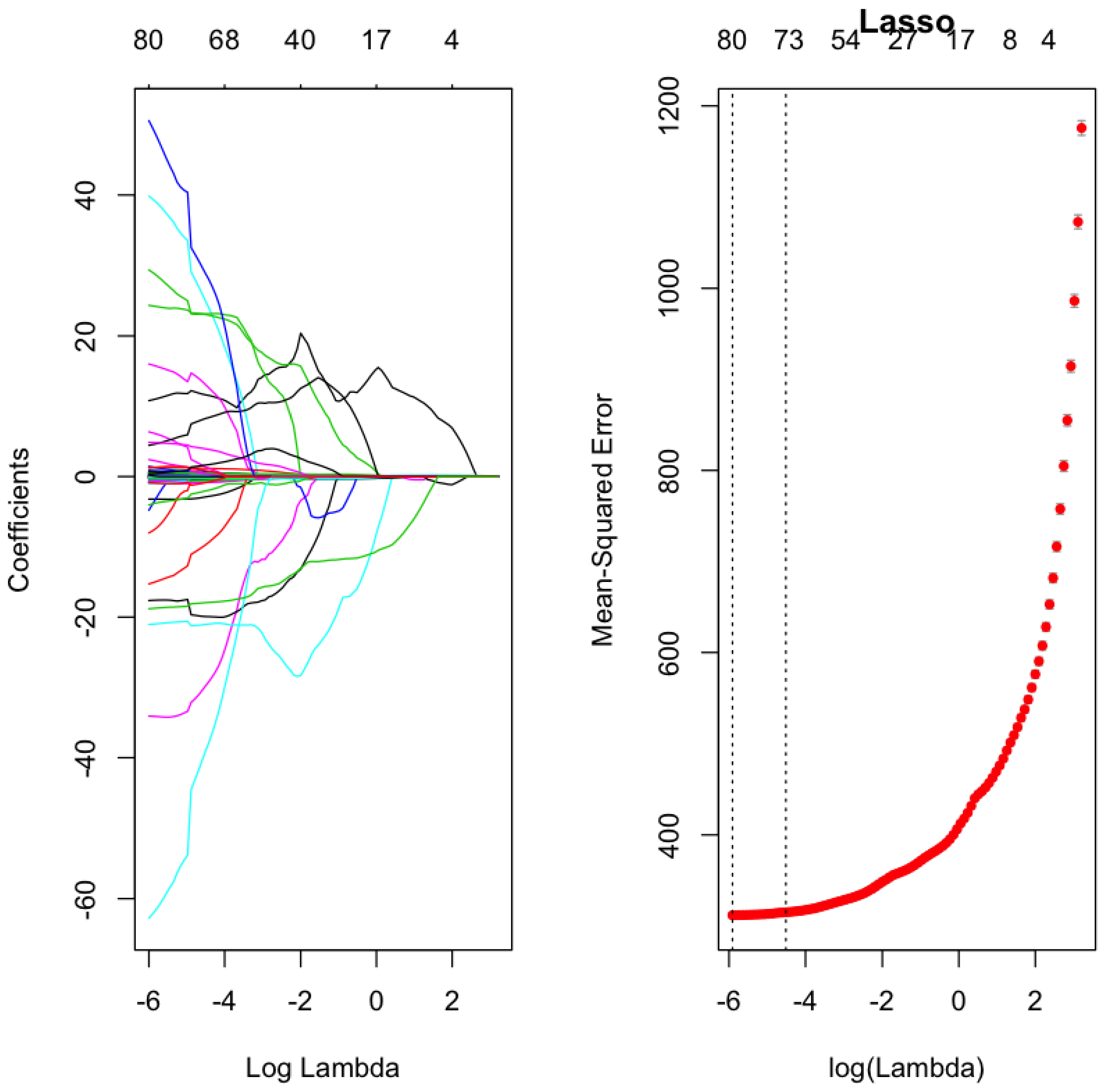
Parameters for **24 features** are shrinked very close to zero. Moreover, 7 features has been completely removed,
i.e. `wtd_gmean_atomic_mass` , `wtd_range_atomic_mass` , `gmean_fie` , `wtd_gmean_fie` , `entropy_fie` , `entropy_atomic_radius` , and `wtd_mean_Valence` .

Lasso

Now, we will look at the performace of Lasso Regression.

```
In [44]: # Fitting Lasso regression to the data
         fit.lasso <- glmnet(as.matrix(train.data), as.matrix(train.label), family="gaussian", alpha=1)
         fit.lasso.cv <- cv.glmnet(as.matrix(train.data), as.matrix(train.label), type.measure="mse", alpha=1,
                                   family="gaussian")
```

```
In [45]: # Plot solution paths:
         par(mfrow=c(1,2))
         plot(fit.lasso,xvar="lambda")
         plot(fit.lasso.cv, main="Lasso")
```



We will use `lambda.1se` to get the parsimonious model, since 'lambda.min' retruns the lambda with least RMSE, achieved on the expense of adding more features to our model

```
In [46]: test_pred <- predict(fit.lasso.cv, s=fit.lasso.cv$lambda.1se, newx=as.matrix(test.data))

         # Checking MSE
         mse_lasso <- sqrt(mean((test.label - test_pred)^2))
         print(paste('RMSE for Lasso:',mse_lasso))

         # Checking R-Squared Value
         rsq_lasso <- cor(test.label, test_pred)^2
         print(paste('R-Squared for Lasso:',rsq_lasso))

         [1] "RMSE for Lasso: 17.8312560227451"
         [1] "R-Squared for Lasso: 0.727342310788728"
```

In [47]:

```
data.frame(predict(fit.lasso.cv, s = fit.lasso.cv$lambda.1se, type = "coefficients")[1:82,]) -> features.lasso
names(features.lasso) <- 'weight'
Features <- row.names(features.lasso)
row.names(features.lasso) <- 1:nrow(features.lasso)
features.lasso<-cbind(Features,features.lasso)
# Check the features with zero weights
features.lasso[(abs(features.lasso$weight)< 0.05),]

#storing values
model_comp[3,] <- c('Lasso', round(rsq_lasso,3),round(mse_lasso,3),nrow(features.lasso[(abs(features.lasso$weight)!=0),])-1)

# storing important features for Lasso
lasso_model_features <- features.lasso[(abs(features.lasso$weight)!=0),'Features'][-1]
```

	Features	weight
5	gmean_atomic_mass	-2.046120e-02
6	wtd_gmean_atomic_mass	0.000000e+00
10	wtd_range_atomic_mass	0.000000e+00
13	mean_fie	2.305641e-03
14	wtd_mean_fie	1.810265e-02
15	gmean_fie	0.000000e+00
16	wtd_gmean_fie	0.000000e+00
17	entropy_fie	0.000000e+00
20	wtd_range_fie	1.481348e-02
22	wtd_std_fie	-2.416209e-02
27	entropy_atomic_radius	0.000000e+00
33	mean_Density	-3.019671e-03
34	wtd_mean_Density	-1.611216e-04
35	gmean_Density	1.480402e-04
36	wtd_gmean_Density	1.789008e-03
39	range_Density	-1.068902e-03
40	wtd_range_Density	9.429171e-05
41	std_Density	3.846387e-03
42	wtd_std_Density	-1.283754e-03
55	gmean_FusionHeat	-2.588136e-02
56	wtd_gmean_FusionHeat	1.050170e-04
63	mean_ThermalConductivity	-2.515414e-03
69	range_ThermalConductivity	-4.082662e-02
74	wtd_mean_Valence	0.000000e+00

Parameters for **25 features** are shrinked very close to zero. Moreover, 9 features has been completely removed, i.e. `wtd_gmean_atomic_mass` , `wtd_range_atomic_mass` , `gmean_fie` , `wtd_gmean_fie` , `entropy_fie` , `entropy_atomic_radius` , `wtd_gmean_FusionHeat` , `mean_ThermalConductivity` , and `wtd_mean_Valence` .

XGBoost Model

XGBoost has recently been popularly used around the world, to obtain high accuracy models. It is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. Let's fit it to our dataset and check it's performance.

In [48]:

```
# Setting defaullt parameters
params <- list(eta=0.3, gamma=0, max_depth=6, min_child_weight=1, subsample=1, colsample_bytree=1)
```

In [49]:

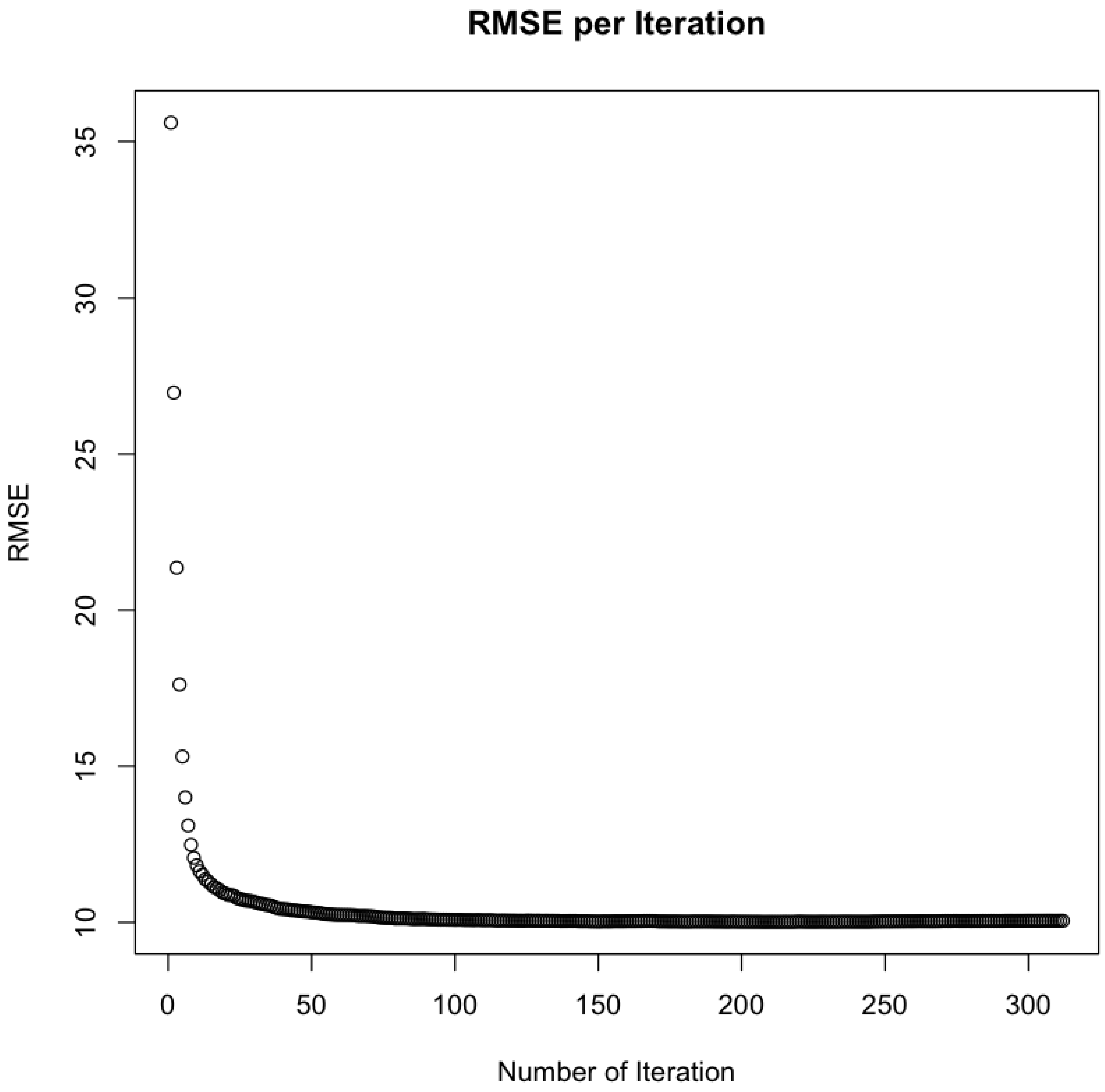
```
 # Training the model using default parameters and more iterations
fit.XGB <- xgb.train(params = params,
                           data = xgb.DMatrix(data = as.matrix(train.data), label = as.matrix(train.label)),
                           nrounds = 3000,
                           nfold = 10,
                           showsd = T,
                           stratified = T,
                           print_every_n = 10,
                           early_stopping_rounds = 100,
                           watchlist = list(test = xgb.DMatrix(data = as.matrix(test.data),label = as.matrix(test.label))))
```

```
[1]	test-rmse:35.607552
Will train until test_rmse hasn't improved in 100 rounds.

[11]	test-rmse:11.639416
[21]	test-rmse:10.880176
[31]	test-rmse:10.629022
[41]	test-rmse:10.420289
[51]	test-rmse:10.315037
[61]	test-rmse:10.236148
[71]	test-rmse:10.193866
[81]	test-rmse:10.125165
[91]	test-rmse:10.098318
[101]	test-rmse:10.082262
[111]	test-rmse:10.068814
[121]	test-rmse:10.054916
[131]	test-rmse:10.048759
[141]	test-rmse:10.042699
[151]	test-rmse:10.027145
[161]	test-rmse:10.037346
[171]	test-rmse:10.030608
[181]	test-rmse:10.023036
[191]	test-rmse:10.023011
[201]	test-rmse:10.019464
[211]	test-rmse:10.014511
[221]	test-rmse:10.021206
[231]	test-rmse:10.016566
[241]	test-rmse:10.019551
[251]	test-rmse:10.029482
[261]	test-rmse:10.035264
[271]	test-rmse:10.043022
[281]	test-rmse:10.041998
[291]	test-rmse:10.044185
[301]	test-rmse:10.048757
[311]	test-rmse:10.050773
Stopping. Best iteration:
[212]	test-rmse:10.013698
```

Let's look at then change in RMSE values with each iteration, and top 10 features given by XGBoost model based on information gain criteria.

In [50]:

```
# Training and test error plots
XGB.log <- data.frame(fit.XGB$evaluation_log)
plot(XGB.log$iter, XGB.log$test_rmse, xlab = 'Number of Iteration', ylab = 'RMSE', main = 'RMSE per Iteration')
```

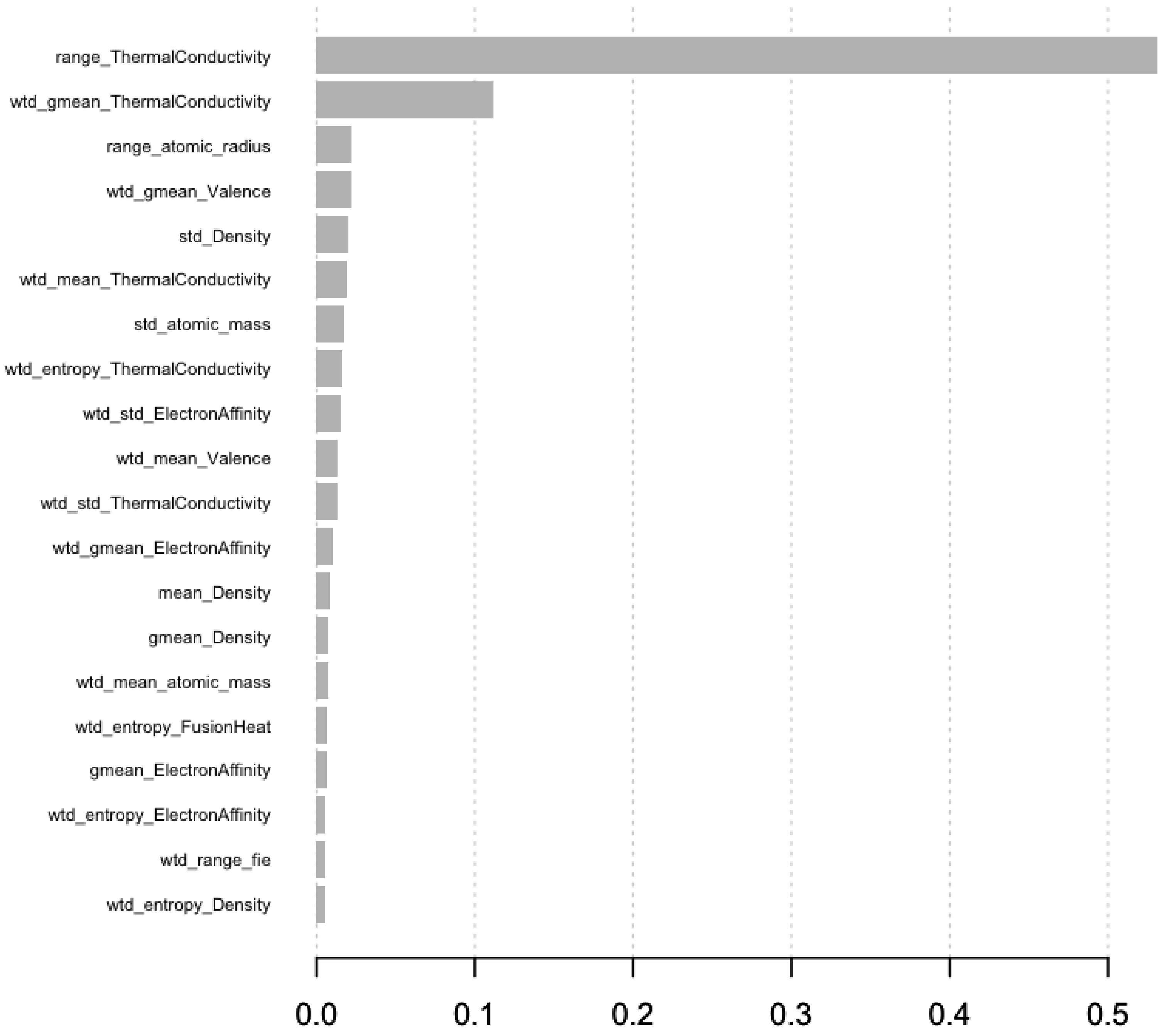


In [51]:

```
# Top 20 Feature Importance
important <- xgb.importance(colnames(train.data), model = fit.XGB)[1:20,]
XGboost_features <- xgb.importance(colnames(train.data), model = fit.XGB)[1:20,'Feature'] # storing those feature
print(important)
```

```
                              Feature        Gain       Cover   Frequency
 1:        range_ThermalConductivity 0.530858650 0.003402466 0.001721170
 2:    wtd_gmean_ThermalConductivity 0.111563588 0.010690532 0.012572027
 3:              range_atomic_radius 0.022404007 0.002807958 0.002694006
 4:                  wtd_gmean_Valence 0.022032106 0.019299672 0.009952855
 5:                      std_Density 0.020389413 0.006754594 0.006136347
 6:     wtd_mean_ThermalConductivity 0.019185678 0.025419760 0.015864701
 7:                  std_atomic_mass 0.017348990 0.005773859 0.006959515
 8: wtd_entropy_ThermalConductivity 0.015958994 0.025003159 0.021701714
 9:         wtd_std_ElectronAffinity 0.015077088 0.020207198 0.015640201
10:                 wtd_mean_Valence 0.013556121 0.019224560 0.012497194
11:      wtd_std_ThermalConductivity 0.013329777 0.046433950 0.025593055
12:       wtd_gmean_ElectronAffinity 0.010119655 0.023122366 0.014293198
13:                     mean_Density 0.008170929 0.005260028 0.006959515
14:                    gmean_Density 0.007667904 0.003779465 0.004564843
15:              wtd_mean_atomic_mass 0.006979937 0.020055465 0.055526454
16:            wtd_entropy_FusionHeat 0.006602387 0.035861724 0.020579211
17:           gmean_ElectronAffinity 0.006546299 0.007159599 0.005163511
18:     wtd_entropy_ElectronAffinity 0.005647591 0.022198289 0.018932874
19:                    wtd_range_fie 0.005443815 0.020169713 0.018858041
20:              wtd_entropy_Density 0.005253146 0.019483935 0.019082541
```

```
In [52]: # Understanding importance with a plot
         xgb.plot.importance(important)
```



`range_ThermalConductivity` seems to be the most important feature selected by the XGBoost algorithm used. Moreover `wtd_gmean_ThermalConductivity` also seems to be quite important compared to rest of the top 18 features. Now, we will evaluate the performance of XGboost model on test set.

```
In [53]: # Test the model on test data
         test_xgbDMatrix <- xgb.DMatrix(data = as.matrix(test.data), label = test.label)
         test_pred <- predict(fit.XGB, newdata = test_xgbDMatrix)

         # Minimum RMSE
         rmse_XGB <- min(XGB.log$test_rmse)
         print(paste('Lowest RMSE with XGBoost :',rmse_XGB))

         # Checking R-Squared Value
         rsq_XGB <- cor(test.label, test_pred)^2
         print(paste('R-Squared for XGB :',rsq_XGB))

         # Storing values
         model_comp[4,] <- c('XGBoost', round(rsq_XGB,3),round(rmse_XGB,3),fit.XGB$nfeatures)
```

```
[1] "Lowest RMSE with XGBoost : 10.013698"
[1] "R-Squared for XGB : 0.91419587337505"
```

Highly accurate predictions with R-squared value of whopping `0.91` is obtained!

4. Model Comparsion

```
In [54]: model_comp
```

Model	R.Squared	R.M.S.E	Features
Linear model + CFS + MRMR	0.677	19.397	28
Elastic Net	0.726	17.887	71
Lasso	0.727	17.831	74
XGBoost	0.914	10.014	81

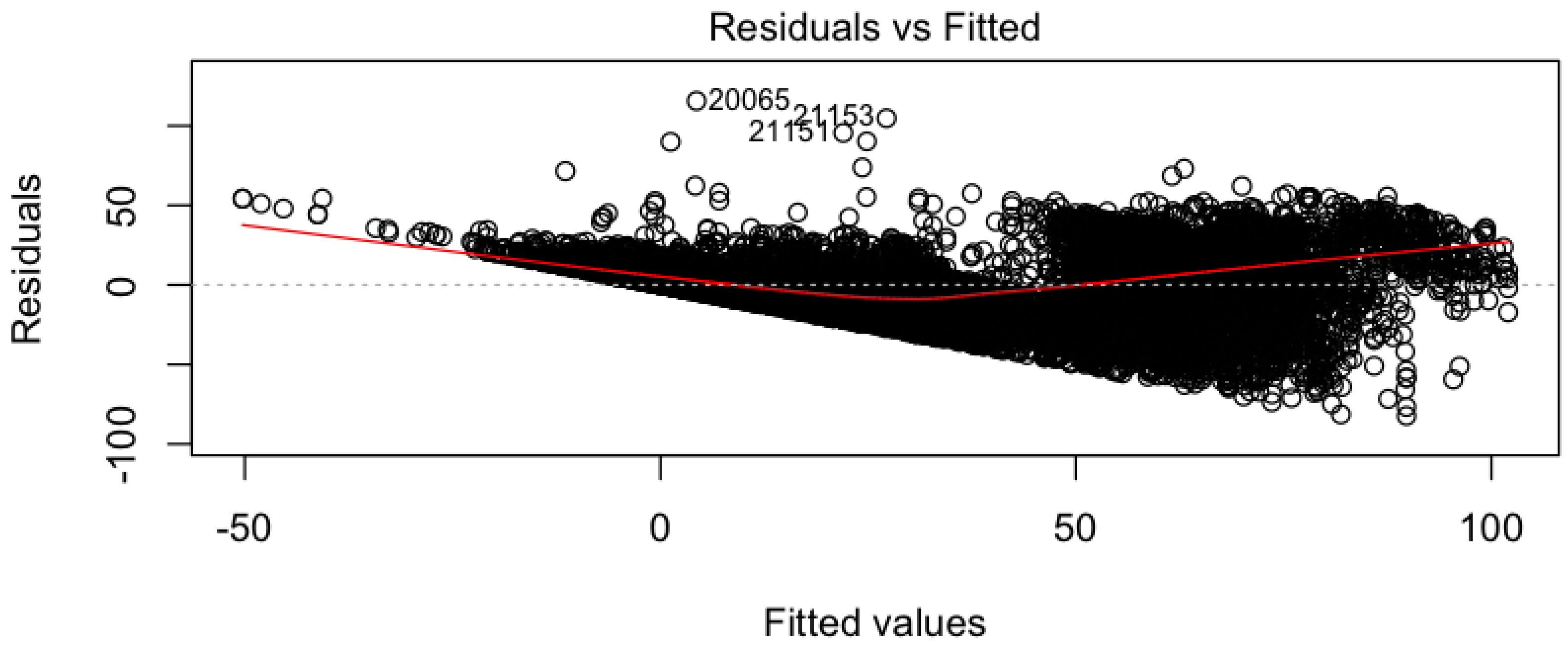
It is observable that `XGBoost` model returns the best fit (R-Squared of 0.91) and least errors (R.M.S.E - 10.01), but a major drawback is high number of features. This makes the model highly complex and difficult to explain. On the lower end of the scale of fit and error is `Linear model with C.F.S and M.R.M.R features (LMCM)` with the R-Squared of 0.68 and RMSE of 19.397. However, a data scientist may argue it's difference with XGBoost model is **compensated** by the vast difference in number of features. i.e. 81-28=53 features.

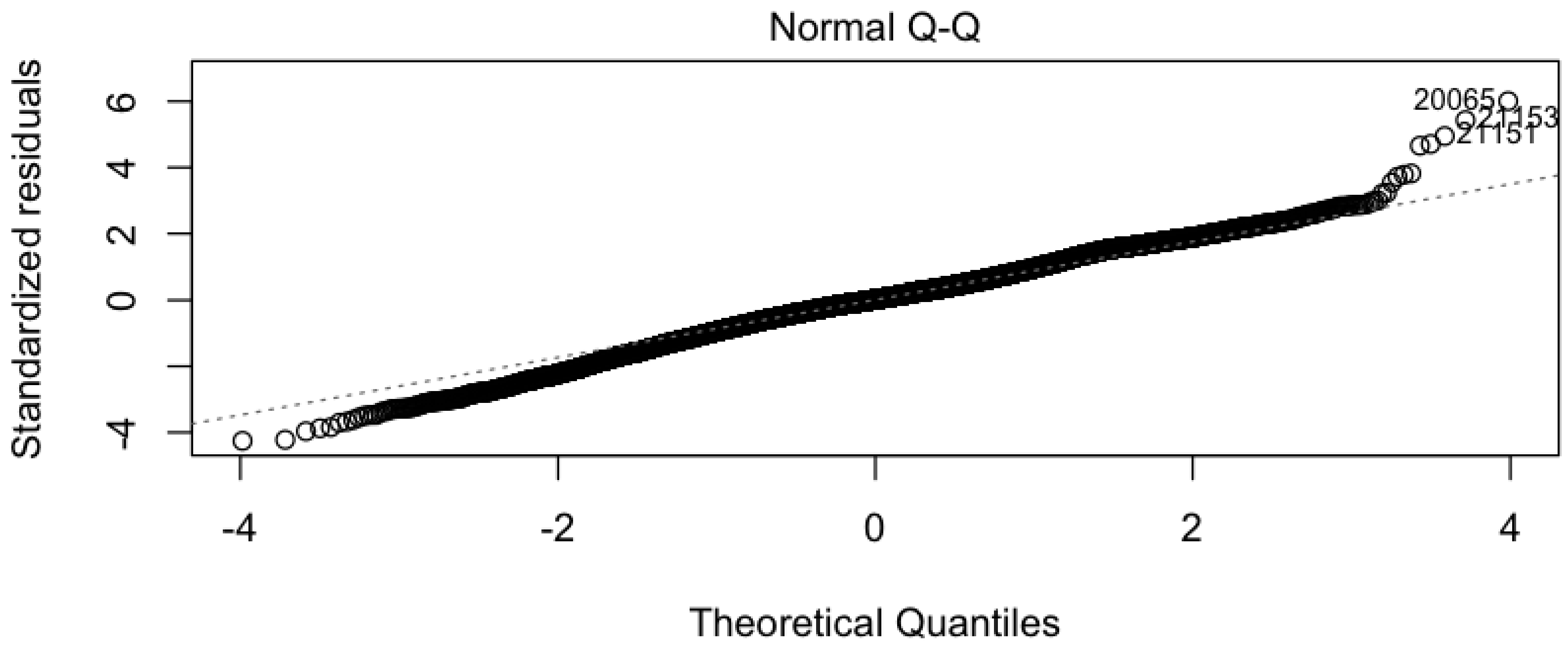
Linear regression with Regularization, namely, `Elastic Net` and `Lasso` , are in middle of the aforementioned models, both, performance and complexity wise. They both have similar R-Squared on test set, i.e. 0.72, and Elastic Net performing slightly better in terms of RMSE. The main difference between these two lies between the number of features, where `Lasso` model has 8 lesser number of features than `Elastic Net` . Thus, it can be concluded that `Lasso` returns **overall better performance** on the superconductor dataset than `Elastic Net` .

The difference among `LMCM` and Regularization methods, isn't very significant in terms of fit anf RMSE. However, This minute difference is overshadowed by the high difference in the number of features.

Let's check the residuals of the linear model created at Model-1.

```
In [55]: par(mfcol=c(2,1))
         plot(fit.step,which=1)
         plot(fit.step,which=2)
```





The diagnostic plots show residuals in four different ways.

1. The **residual vs fitted plot**: We can identify some non-linear trends in the plot, which indicates some information is yet to be captured by our model.
2. The normal **Q-Q plot**: We can observe that most of residuals are parrallel and lined up on the dashed line, which indicates the residuals are normally distributed.

Here, we can conclude that though model is explainable with less number of features and the normally distributed residuals, model still has some more information to extract out of the dataset, confirmed by the non-linear trend observed in residual vs fitted plot.

5. Variable Identification and Explanation

Let's look at the common features selected by the all three models.

```
In [56]: # Priniting important common features in all models
         as.matrix(intersect(intersect(model_1_features,lasso_model_features),lasso_model_features))
```

```
       wtd_gmean_atomic_radius
        mean_ElectronAffinity
             std_atomic_radius
                 range_Valence
              mean_atomic_mass
             wtd_gmean_Density
           entropy_atomic_mass
   wtd_entropy_ElectronAffinity
                   std_Density
                   std_Valence
           gmean_atomic_radius
 wtd_gmean_ThermalConductivity
         wtd_entropy_FusionHeat
             range_atomic_mass
     wtd_range_ElectronAffinity
            range_atomic_radius
 wtd_range_ThermalConductivity
        wtd_entropy_atomic_mass
      mean_ThermalConductivity
                 gmean_Density
           wtd_range_FusionHeat
   wtd_std_ThermalConductivity
            wtd_entropy_Valence
     wtd_gmean_ElectronAffinity
     range_ThermalConductivity
                entropy_Density
```

Following are the most important features selected by all three models:

- **Thermal Conductivity** : 'wtd_gmean_ThermalConductivity', 'wtd_range_ThermalConductivity', 'mean_ThermalConductivity','wtd_std_ThermalConductivity', 'range_ThermalConductivity'
- **Atomic Mass** : 'mean_atomic_mass', 'entropy_atomic_mass', 'range_atomic_mass', 'wtd_entropy_atomic_mass'
- **Density** : 'wtd_gmean_Density', 'std_Density', 'range_atomic_radius', 'gmean_Density', 'entropy_Density'
- **Atomic Radius** : 'wtd_gmean_atomic_radius', 'std_atomic_radius', 'gmean_atomic_radius'
- **Valence** : 'range_Valence', 'std_Valence', 'wtd_entropy_Valence'
- **Electron Affinity** : 'mean_ElectronAffinity', 'wtd_entropy_ElectronAffinity' ,'wtd_range_ElectronAffinity', 'wtd_gmean_ElectronAffinity'
- **Fusion Heat** : 'wtd_entropy_FusionHeat', 'wtd_range_FusionHeat'

It seems that `Thermal Conductivity` is the most important property, since it has most number of descriptions selected as features in our models, followed by `Density` , `Atomic Radius` and `Atomic Mass` . `First Ionization Energy` seems to have least effect on superconductivity, since no model selected as an important feature.

Upon, researching more about `Superconductivity` , it can be confirmed that `Thermal Conductivity` has high positive correlation with superconductivity, since they both measure similar property of any element, i.e., conducive characteristic of element.

6. Conclusion

In the industry of Data Science, where 'No model is correct, and some are useful', the choice of model is subjective and dynamic to business needs. It can be left upon Data Scientist to choose which 'model is useful' for requirements of the Data Science project. Both, a non-parasimonious model with high accuracy , and a parsimonious model with releatively lower accuracy are developed. First one can be used, if the objective of the project is `Predictive Analysis` , where only getting highly accurate results matters, and latter can be used if the objective `Descriptive Analysis` , where power of explanations matters.

In this project, we built various models with ranging accuracy on the test set. `XGBoost` model was found to be performing the best on the 'Superconductor' data set, at the expense of high number of features. However, the most parsimonious linear model developed is built by using combination of, `MRMR Feature selection` , `Correlation based Feature Selection` , and `Hybrid Information Gain Selction` techniques.

From our analysis, it can be said that `Thermal Conductivity` is the most important property of elements, followed by `Density` , `Atomic Radius` and `Atomic Mass` . `First Ionization Energy` appears to have the least or no effect. Physicists can use the results from our analysis, and identify factors affecting superconductivity, and probably find a way to quantitaively measure this property down the road!

7. References

- Beginners Tutorial on XGBoost and Parameter Tuning in R Tutorials & Notes | Machine Learning | HackerEarth. (2019). Retrieved 15 September 2019, from https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/beginners-tutorial-on-xgboost-parameter-tuning-r/tutorial/ (https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/beginners-tutorial-on-xgboost-parameter-tuning-r/tutorial/)
- MRMR function | R Documentation. (2019). Retrieved 15 September 2019, from https://www.rdocumentation.org/packages/praznik/versions/6.0.0/topics/MRMR (https://www.rdocumentation.org/packages/praznik/versions/6.0.0/topics/MRMR)
- R, M. (2019). Multicollinearity Essentials and VIF in R - Articles - STHDA. Retrieved 15 September 2019, from http://www.sthda.com/english/articles/39-regression-model-diagnostics/160-multicollinearity-essentials-and-vif-in-r/ (http://www.sthda.com/english/articles/39-regression-model-diagnostics/160-multicollinearity-essentials-and-vif-in-r/)
- Haqqani, M. (2019). FIT5149 Tutorial Week 1-6 [Ebook]. Grayming Liu.