FINAL PROJECT ML

2023-04-15

EXPLORATORY DATA ANALYSIS

```r
ft_fire <- read.csv("forestfires.csv",header=T)
data("f_fire")
```

```
## Warning in data("f_fire"): data set 'f_fire' not found
```

```r
str(ft_fire)
```

```
## 'data.frame':    517 obs. of  13 variables:
##  $ X    : int  7 7 7 8 8 8 8 8 8 7 ...
##  $ Y    : int  5 4 4 6 6 6 6 6 6 5 ...
##  $ month: chr  "mar" "oct" "oct" "mar" ...
##  $ day  : chr  "fri" "tue" "sat" "fri" ...
##  $ FFMC : num  86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...
##  $ DMC  : num  26.2 35.4 43.7 33.3 51.3 ...
##  $ DC   : num  94.3 669.1 686.9 77.5 102.2 ...
##  $ ISI  : num  5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...
##  $ temp : num  8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...
##  $ RH   : int  51 33 33 97 99 29 27 86 63 40 ...
##  $ wind : num  6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...
##  $ rain : num  0 0 0 0.2 0 0 0 0 0 0 ...
##  $ area : num  0 0 0 0 0 0 0 0 0 0 ...
```

```r
summary(ft_fire)
```

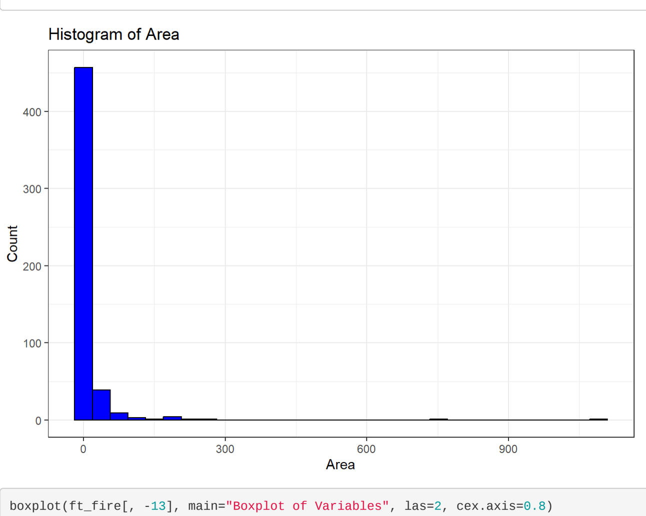
```
##        X               Y          month               day
##  Min.   :1.000   Min.   :2.0   Length:517         Length:517
##  1st Qu.:3.000   1st Qu.:4.0   Class :character   Class :character
##  Median :4.000   Median :4.0   Mode  :character   Mode  :character
##  Mean   :4.669   Mean   :4.3
##  3rd Qu.:7.000   3rd Qu.:5.0
##  Max.   :9.000   Max.   :9.0
##       FFMC            DMC              DC             ISI
##  Min.   :18.70   Min.   :  1.1   Min.   :  7.9   Min.   : 0.000
##  1st Qu.:90.20   1st Qu.: 68.6   1st Qu.:437.7   1st Qu.: 6.500
##  Median :91.60   Median :108.3   Median :664.2   Median : 8.400
##  Mean   :90.64   Mean   :110.9   Mean   :547.9   Mean   : 9.022
##  3rd Qu.:92.90   3rd Qu.:142.4   3rd Qu.:713.9   3rd Qu.:10.800
##  Max.   :96.20   Max.   :291.3   Max.   :860.6   Max.   :56.100
##       temp             RH              wind            rain
##  Min.   : 2.20   Min.   : 15.00   Min.   :0.400   Min.   :0.00000
##  1st Qu.:15.50   1st Qu.: 33.00   1st Qu.:2.700   1st Qu.:0.00000
##  Median :19.30   Median : 42.00   Median :4.000   Median :0.00000
##  Mean   :18.89   Mean   : 44.29   Mean   :4.018   Mean   :0.02166
##  3rd Qu.:22.80   3rd Qu.: 53.00   3rd Qu.:4.900   3rd Qu.:0.00000
##  Max.   :33.30   Max.   :100.00   Max.   :9.400   Max.   :6.40000
##       area
##  Min.   :   0.00
##  1st Qu.:   0.00
##  Median :   0.52
##  Mean   :  12.85
##  3rd Qu.:   6.57
##  Max.   :1090.84
```

```r
head(ft_fire)
```

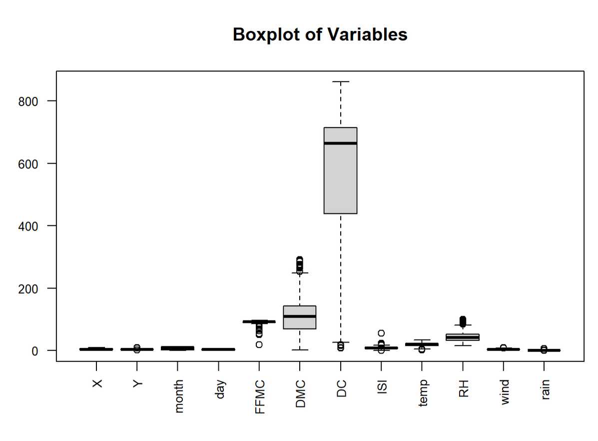
```
##   X Y month day FFMC  DMC    DC  ISI temp RH wind rain area
## 1 7 5   mar fri 86.2 26.2  94.3  5.1  8.2 51  6.7  0.0    0
## 2 7 4   oct tue 90.6 35.4 669.1  6.7 18.0 33  0.9  0.0    0
## 3 7 4   oct sat 90.6 43.7 686.9  6.7 14.6 33  1.3  0.0    0
## 4 8 6   mar fri 91.7 33.3  77.5  9.0  8.3 97  4.0  0.2    0
## 5 8 6   mar sun 89.3 51.3 102.2  9.6 11.4 99  1.8  0.0    0
## 6 8 6   aug sun 92.3 85.3 488.0 14.7 22.2 29  5.4  0.0    0
```

```r
ft_fire$month <- as.numeric(factor(ft_fire$month))
ft_fire$day <- as.numeric(factor(ft_fire$day))
ggplot(data = ft_fire, aes(x = area)) +
  geom_histogram(fill = "blue", color = "black") +
  labs(title = "Histogram of Area", x = "Area", y = "Count") +
  theme_bw()
```

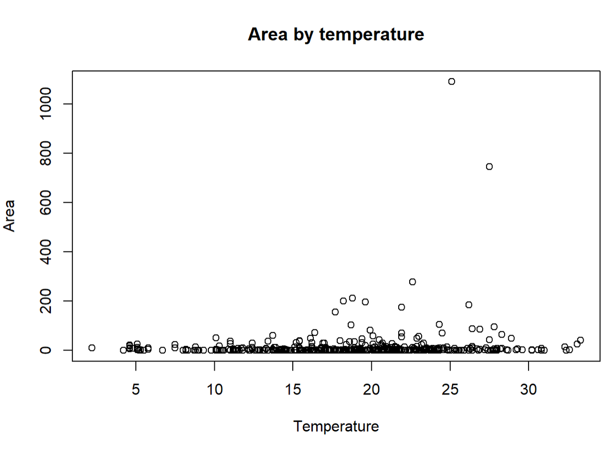
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



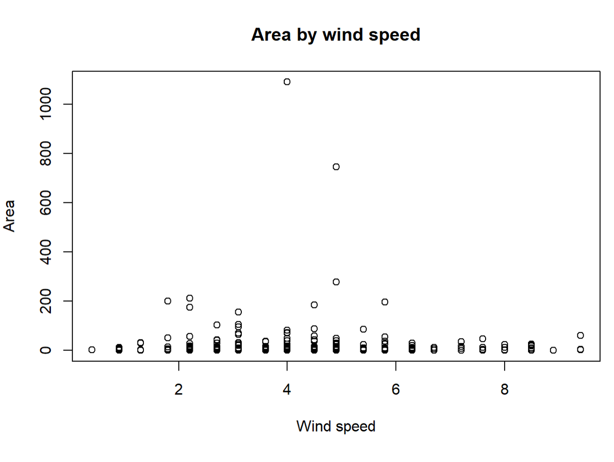

```r
boxplot(ft_fire[, -13], main="Boxplot of Variables", las=2, cex.axis=0.8)
```



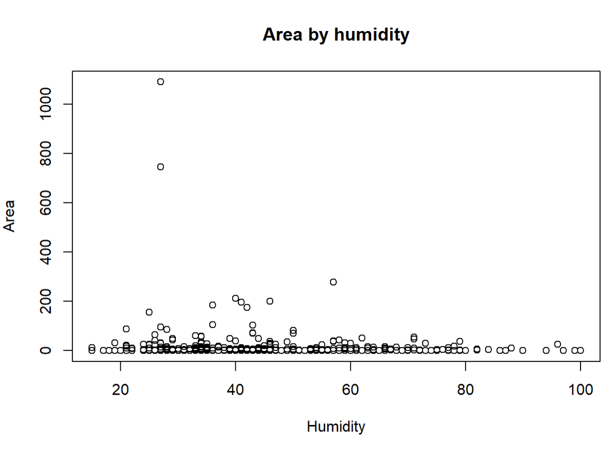

```r
plot(area ~ temp, data = ft_fire, main = "Area by temperature", xlab = "Temperature", ylab = "Area")
```




```r
plot(area ~ wind, data = ft_fire, main = "Area by wind speed", xlab = "Wind speed", ylab = "Area")
```




```r
plot(area ~ RH, data = ft_fire, main = "Area by humidity", xlab = "Humidity", ylab = "Area")
```




```r
tail(ft_fire)
```

```
##     X Y month day FFMC   DMC    DC  ISI temp RH wind rain  area
## 512 8 6     2   4 81.6  56.7 665.6  1.9 27.8 35  2.7    0  0.00
## 513 4 3     2   4 81.6  56.7 665.6  1.9 27.8 32  2.7    0  6.44
## 514 2 4     2   4 81.6  56.7 665.6  1.9 21.9 71  5.8    0 54.29
## 515 7 4     2   4 81.6  56.7 665.6  1.9 21.2 70  6.7    0 11.16
## 516 1 4     2   3 94.4 146.0 614.7 11.3 25.6 42  4.0    0  0.00
## 517 6 3    10   6 79.5   3.0 106.7  1.1 11.8 31  4.5    0  0.00
```

DATA PREPROCESSING

```r
set.seed(123)

 Q1 <- quantile(ft_fire$area, 0.25)
 Q3 <- quantile(ft_fire$area, 0.75)
 IQR <- Q3 - Q1
 ft_fire <- ft_fire[ft_fire$area <= Q3 + 1.5*IQR, ]
ft_fire$log_area <- log(ft_fire$area + 1)

scaled_predictors <- scale(ft_fire[, -14])


fire_nmm <- cbind(scaled_predictors, log_area = as.data.frame(ft_fire$log_area))


fire_nmm$log_area <- ft_fire$log_area


fire_nmm <- fire_nmm[, -which(names(fire_nmm) == "area")]
fire_nmm <- fire_nmm[, -which(names(fire_nmm) == "ft_fire$log_area")]
fire_nmm <- fire_nmm[, -which(names(fire_nmm) == "month")]
fire_nmm <- fire_nmm[, -which(names(fire_nmm) == "day")]

tail(fire_nmm)
```

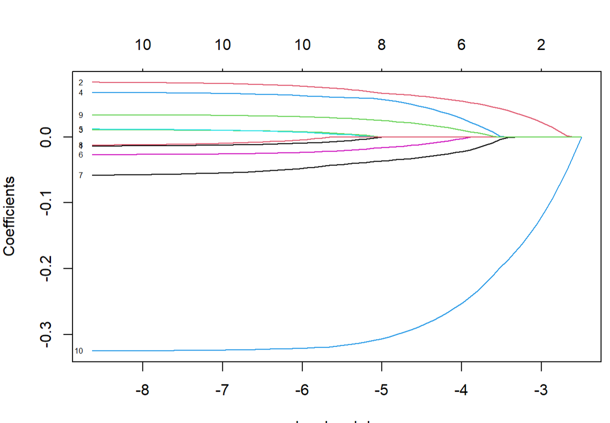
```
##              X          Y        FFMC        DMC         DC        ISI
## 511  0.5991227  0.5650757  0.07372876  0.8822143  0.8293628 -0.4194354
## 512  1.4667518  1.3743639 -1.55150292 -0.8398904  0.4806821 -1.5422418
## 513 -0.2685064 -1.0535007 -1.55150292 -0.8398904  0.4806821 -1.5422418
## 515  1.0329373 -0.2442125 -1.55150292 -0.8398904  0.4806821 -1.5422418
## 516 -1.5699500 -0.2442125  0.66157852  0.5556083  0.2766838  0.4874468
## 517  0.5991227 -1.0535007 -1.91458659 -1.6790649 -1.7592911 -1.7149813
##           temp         RH        wind        rain  log_area
## 511 -0.1153148  1.0666983  0.80448092 -0.07812976 0.3576744
## 512  1.5644580 -0.5920088 -0.71404805 -0.07812976 0.0000000
## 513  1.5644580 -0.7763096 -0.71404805 -0.07812976 2.0068708
## 515  0.4096142  1.5581671  1.53562449 -0.07812976 2.4981519
## 516  1.1795101 -0.1619736  0.01709553 -0.07812976 0.0000000
## 517 -1.2351633 -0.8377432  0.29830460 -0.07812976 0.0000000
```

LASSO REGRESSION

```r
set.seed(123)
train_index <- createDataPartition(fire_nmm$log_area, p = 0.7, list = FALSE)
train <- fire_nmm[train_index, ]
test <- fire_nmm[-train_index, ]

xtrain <- train[, -11]
ytrain <- train$log_area
xtest <- test[, -11]
ytest <- test$log_area



lasso <- glmnet(xtrain, ytrain, alpha=1)
cvv <- cv.glmnet(as.matrix(xtrain), ytrain, alpha=1)
bestlambda <- cvv$lambda.min
print(paste0("Optimal lambda: ", bestlambda))
```

```
## [1] "Optimal lambda: 0.0567552491975546"
```

```r
lasso_best <- glmnet(xtrain, ytrain, alpha=1, lambda=bestlambda)


plot(lasso, xvar = "lambda", label=T)
```




```r
# Evaluate the model
ypred <- predict(lasso_best, as.matrix(xtest))

ypred <- exp(ypred) - 1
mse <- mean( ((exp(ytest) - 1) - ypred)  ^2)
RMSE.lasso <- sqrt(mse)
MAD.lasso <- mean(abs(((exp(ytest)-1)) - ypred))

print(paste0("Optimal lambda: ", bestlambda))
```

```
## [1] "Optimal lambda: 0.0567552491975546"
```

```r
print(paste0("RMSE: ", RMSE.lasso))
```

```
## [1] "RMSE: 3.62129748725139"
```

```r
print(paste0("MSE: ", mse))
```

```
## [1] "MSE: 13.1137954911732"
```

```r
print(paste0("MAD: ", MAD.lasso))
```

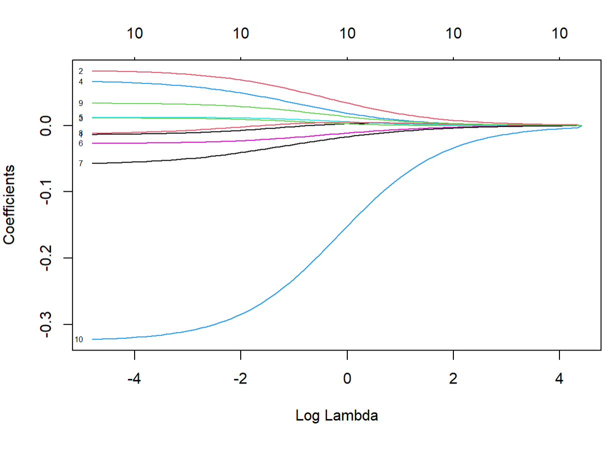
```
## [1] "MAD: 2.28599996826835"
```

RIDGE REGRESSION

```r
set.seed(123)
train_indexx <- createDataPartition(fire_nmm$log_area, p = 0.7, list = FALSE)
train_data <- fire_nmm[train_indexx, ]
test_data <- fire_nmm[-train_indexx, ]

x_train <- train_data[, -11]
y_train <- train_data$log_area
x_test <- test_data[, -11]
y_test <- test_data$log_area



ridge <- glmnet(x_train, y_train, alpha=0)
cvfit <- cv.glmnet(as.matrix(x_train), y_train, alpha=0)
lambdaa <- cvfit$lambda.min


print(paste0("Optimal lambda: ", lambdaa))
```

```
## [1] "Optimal lambda: 82.3421488665209"
```

```r
ridge_best <- glmnet(x_train, y_train, alpha=0, lambda=lambdaa)


plot(ridge, xvar = "lambda", label=T)
```




```r
# Evaluate the model
ypredd <- predict(ridge_best, as.matrix(x_test), s= lambdaa)

ypredd <- exp(ypredd) - 1

mse_r <- mean((((exp(y_test)-1)) - ypredd)^2)
RMSE.ridge <- sqrt(mse_r)
MAD.ridge <- mean(abs(((exp(y_test)-1)) - ypredd))


print(paste0("Optimal lambda: ", lambdaa))
```

```
## [1] "Optimal lambda: 82.3421488665209"
```

```r
print(paste0("MSE: ", mse_r))
```

```
## [1] "MSE: 12.8652853331844"
```

```r
print(paste0("RMSE: ", RMSE.ridge))
```

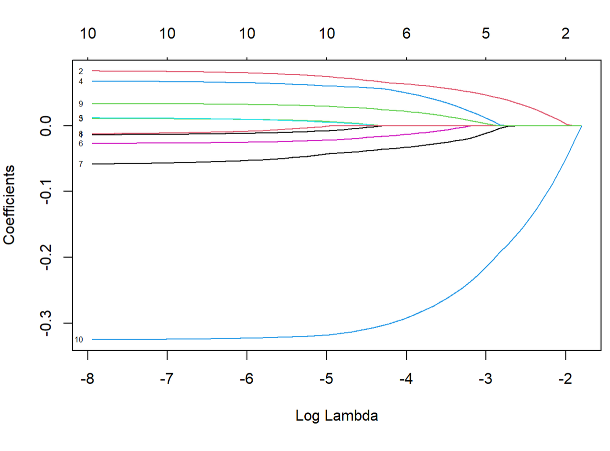
```
## [1] "RMSE: 3.58682106233144"
```

```r
print(paste0("MAD: ", MAD.ridge))
```

```
## [1] "MAD: 2.26842480029263"
```

ELASTIC NET

```r
set.seed(123)
train_index.elas <- createDataPartition(fire_nmm$log_area, p = 0.7, list = FALSE)
train_elas <- fire_nmm[train_index.elas, ]
test_elas <- fire_nmm[-train_index.elas, ]

x_train_elas <- train_elas[, -11]
y_train_elas <- train_elas$log_area
x_test_elas <- test_elas[, -11]
y_test_elas <- test_elas$log_area



elastic <- glmnet(x_train_elas, y_train_elas, alpha=0.5)
cv.elastic <- cv.glmnet(as.matrix(x_train_elas), y_train_elas, alpha=0.5)
lambda.elastic <- cv.elastic$lambda.min




elas_best <- glmnet(x_train_elas, y_train_elas, alpha=0.5, lambda=lambda.elastic)


plot(elastic, xvar = "lambda", label=T)
```




```r
# Evaluate the model
ypred.elas <- predict(elas_best, as.matrix(x_test_elas), s= lambda.elastic)

ypred.elas <- exp(ypred.elas) - 1

elas.mse <- mean((((exp(y_test_elas)-1)) - ypred.elas)^2)
RMSE.elas <- sqrt(elas.mse)
MAD.elas <- mean(abs(((exp(y_test_elas)-1)) - ypred.elas))


print(paste0("Optimal lambda: ", lambda.elastic))
```

```
## [1] "Optimal lambda: 0.113510498395109"
```

```r
print(paste0("MSE: ", elas.mse))
```

```
## [1] "MSE: 13.1080284834"
```

```r
print(paste0("RMSE: ", RMSE.elas))
```

```
## [1] "RMSE: 3.62050113705271"
```

```r
print(paste0("MAD: ", MAD.elas))
```

```
## [1] "MAD: 2.28542549806104"
```

RANDOM FOREST

```r
set.seed(123)
train_ind <- createDataPartition(fire_nmm$log_area, p = 0.7, list = FALSE)
train_ran <- fire_nmm[train_ind, ]
test_ran <- fire_nmm[-train_ind, ]


model_ran <- randomForest(log_area ~ ., data = train_ran)




pred_ran <- predict(model_ran, newdata = test_ran)


pred_ran <- exp(pred_ran) -1

mse_ran <- mean(((exp(test_ran$log_area)-1) - pred_ran)^2)
RMSE.ran <- sqrt(mse_ran)
MAD.ran <- mean(abs((exp(test_ran$log_area)-1) - pred_ran))

print(paste0("MSE: ", mse_ran))
```

```
## [1] "MSE: 12.1263775541386"
```

```r
print(paste0("RMSE: ", RMSE.ran))
```

```
## [1] "RMSE: 3.48229486892461"
```

```r
print(paste0("MAD: ", MAD.ran))
```

```
## [1] "MAD: 2.16273588413039"
```

SUPPORT VECTOR REGRESSION

```r
set.seed(123)
train_ind_svr <- createDataPartition(fire_nmm$log_area, p = 0.7, list = FALSE)
train_svr <- fire_nmm[train_ind_svr, ]
test_svr <- fire_nmm[-train_ind_svr, ]

tail(train_svr)
```

```
##              X          Y        FFMC        DMC        DC        ISI
## 508 -1.1361354 -0.2442125  0.07372876  0.8822143 0.8293628 -0.4194354
## 511  0.5991227  0.5650757  0.07372876  0.8822143 0.8293628 -0.4194354
## 512  1.4667518  1.3743639 -1.55150292 -0.8398904 0.4806821 -1.5422418
## 513 -0.2685064 -1.0535007 -1.55150292 -0.8398904 0.4806821 -1.5422418
## 515  1.0329373 -0.2442125 -1.55150292 -0.8398904 0.4806821 -1.5422418
## 516 -1.5699500 -0.2442125  0.66157852  0.5556083 0.2766838  0.4874468
##           temp         RH        wind        rain  log_area
## 508  1.2320030 -0.2234072 -0.20787172 -0.07812976 0.0000000
## 511 -0.1153148  1.0666983  0.80448092 -0.07812976 0.3576744
## 512  1.5644580 -0.5920088 -0.71404805 -0.07812976 0.0000000
## 513  1.5644580 -0.7763096 -0.71404805 -0.07812976 2.0068708
## 515  0.4096142  1.5581671  1.53562449 -0.07812976 2.4981519
## 516  1.1795101 -0.1619736  0.01709553 -0.07812976 0.0000000
```

```r
tunemodel_svr <- tune(svm,log_area ~ ., data = train_svr, kernel = "radial",  ranges = list(cost = c(0.01, 0.1,
1, 10, 100),
                                epsilon = c(0.1, 0.2, 0.5,0.8, 1)))

print(tunemodel_svr$best.model)
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = log_area ~ ., data = train_svr,
##     ranges = list(cost = c(0.01, 0.1, 1, 10, 100), epsilon = c(0.1,
##         0.2, 0.5, 0.8, 1)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  0.1
##       gamma:  0.1
##     epsilon:  0.8
##
##
## Number of Support Vectors:  184
```

```r
model_svr <- svm(log_area ~ ., data = train_svr, kernel = "radial", cost = 0.1, epsilon = 0.8)


pred_svr <- predict(model_svr, newdata = test_svr)

pred_svr <- exp(pred_svr) -1

mse_svr <- mean(((exp(test_svr$log_area)-1) - pred_svr)^2)
RMSE.svr <- sqrt(mse_svr)
MAD.svr <- mean(abs((exp(test_svr$log_area)-1) - pred_svr))

print(paste0("MSE: ", mse_svr))
```

```
## [1] "MSE: 12.6357233863891"
```

```r
print(paste0("RMSE: ", RMSE.svr))
```

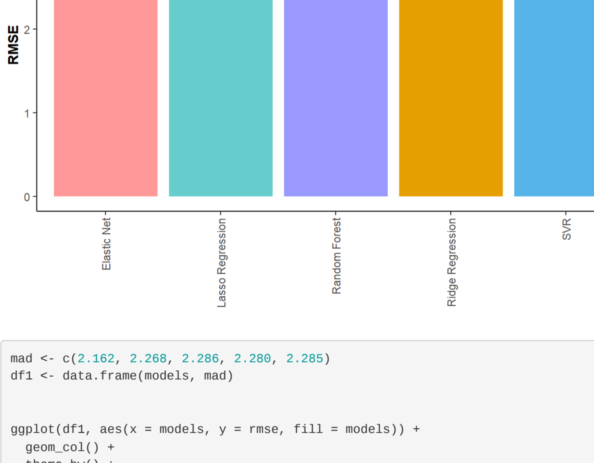
```
## [1] "RMSE: 3.55467627026556"
```

```r
print(paste0("MAD: ", MAD.svr))
```

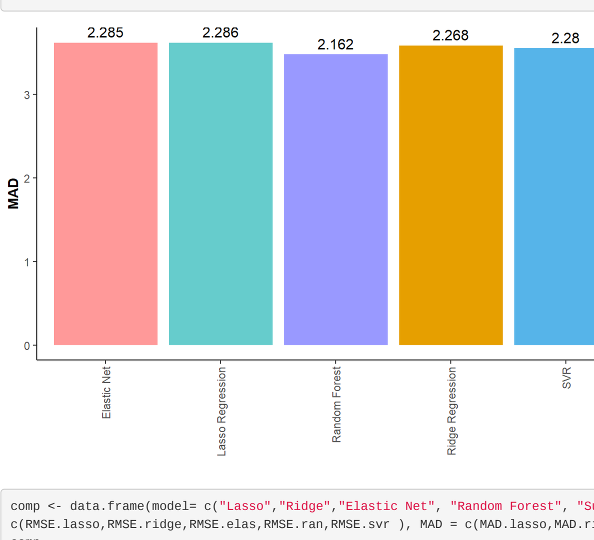
```
## [1] "MAD: 2.28095872496435"
```

SUMMARIZATION

```r
models <- c("Random Forest", "Ridge Regression", "Lasso Regression", "SVR", "Elastic Net")
rmse <- c(3.482, 3.586, 3.621, 3.554, 3.620)
df <- data.frame(models, rmse)


ggplot(df, aes(x = models, y = rmse, fill = models)) +
  geom_col() +
  theme_bw() +
  labs(x = "", y = "RMSE") +
  scale_fill_manual(values = c("#FF9999", "#66CCCC", "#9999FF", "#E69F00", "#56B4E9")) +
  geom_text(aes(label = rmse), vjust = -0.5, color = "black", size = 4) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        legend.position = "none",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        axis.line = element_line(colour = "black"),
        axis.title = element_text(face = "bold"))
```




```r
mad <- c(2.162, 2.268, 2.286, 2.280, 2.285)
df1 <- data.frame(models, mad)


ggplot(df1, aes(x = models, y = rmse, fill = models)) +
  geom_col() +
  theme_bw() +
  labs(x = "", y = "MAD") +
  scale_fill_manual(values = c("#FF9999", "#66CCCC", "#9999FF", "#E69F00", "#56B4E9")) +
  geom_text(aes(label = mad), vjust = -0.5, color = "black", size = 4) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        legend.position = "none",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        axis.line = element_line(colour = "black"),
        axis.title = element_text(face = "bold"))
```




```r
comp <- data.frame(model= c("Lasso","Ridge","Elastic Net", "Random Forest", "Support Vector Regression"), RMSE =
c(RMSE.lasso,RMSE.ridge,RMSE.elas,RMSE.ran,RMSE.svr ), MAD = c(MAD.lasso,MAD.ridge,MAD.elas,MAD.ran,MAD.svr))
comp
```

```
##                       model     RMSE      MAD
## 1                     Lasso 3.621297 2.286000
## 2                     Ridge 3.586821 2.268425
## 3               Elastic Net 3.620501 2.285425
## 4             Random Forest 3.482295 2.162736
## 5 Support Vector Regression 3.554676 2.280959
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.