TP2 MRR

Noah KWA MOUTOME - Victor TAN

2023-10-31

IV. Cookies Study

First, let's go back to our previous results: We saw that with linear regression without penalty, only 2 explanatory variables were significant. We can deduce that we can also use a Lasso regression to select the most important features to predict the fat value of a cookie. Futhermore, we can use a Ridge regression to see if the model is overfitting or not.

Imports

```
cookies_data <- read.csv("cookies.csv")
```

Features extraction

For each line (meaning, for each cookie), we will use the different spectral values to compute: the mean, the standard deviation, the slope, the minimum and the maximum.

```
# Computation (mean, standard deviation, minimum and maximum)

cookies_data$mean <- rowMeans(cookies_data[, 2:701])
cookies_data$stDev <- apply(cookies_data[, 2:701], 1, sd)
cookies_data$min <- apply(cookies_data[, 2:701], 1, min)
cookies_data$max <- apply(cookies_data[, 2:701], 1, max)


# Computation (slope)

# Function: compute_slope
# @param: spectrum_values of a cookie (here, column 2 to 701)
# @return: slope of the spectrum curve for a cookie
compute_slope <- function(spectrum_values) {
  pos <- 1:length(spectrum_values)
  lm_model <- lm(spectrum_values ~ pos)
  slope <- coef(lm_model)[2]
  return(slope)
}

cookies_data$slope <- apply(cookies_data[, 2:701], 1, compute_slope)
```

```r
# Display of the new columns
head(cookies_data[,702:706])
```

```
##        mean     stDev      min     max       slope
## 1 0.9851499 0.4111868 0.259270 1.73946 0.001914311
## 2 1.0355417 0.4123933 0.266864 1.66273 0.001898164
## 3 1.0010620 0.4025158 0.251654 1.60960 0.001860203
## 4 1.0280481 0.4040351 0.277777 1.63881 0.001861782
## 5 1.0655011 0.4158252 0.288328 1.70320 0.001910926
## 6 1.0840236 0.4262425 0.284625 1.74356 0.001967228
```

Regression model

Now, we have the different features of the spectra.

```r
# Only features and fat values are retrieved

cookies_features <- cookies_data[c("fat", "mean", "stDev", "slope", "min", "max")]
head(cookies_features)
```

```
##     fat      mean     stDev       slope      min     max
## 1 12.57 0.9851499 0.4111868 0.001914311 0.259270 1.73946
## 2 15.13 1.0355417 0.4123933 0.001898164 0.266864 1.66273
## 3 12.63 1.0010620 0.4025158 0.001860203 0.251654 1.60960
## 4 13.85 1.0280481 0.4040351 0.001861782 0.277777 1.63881
## 5 15.25 1.0655011 0.4158252 0.001910926 0.288328 1.70320
## 6 13.66 1.0840236 0.4262425 0.001967228 0.284625 1.74356
```

```r
X <- as.matrix(cookies_features[, -1]) # co-variables
y <- cookies_features$fat # target variable
```

```
## Le chargement a nécessité le package : Matrix
```

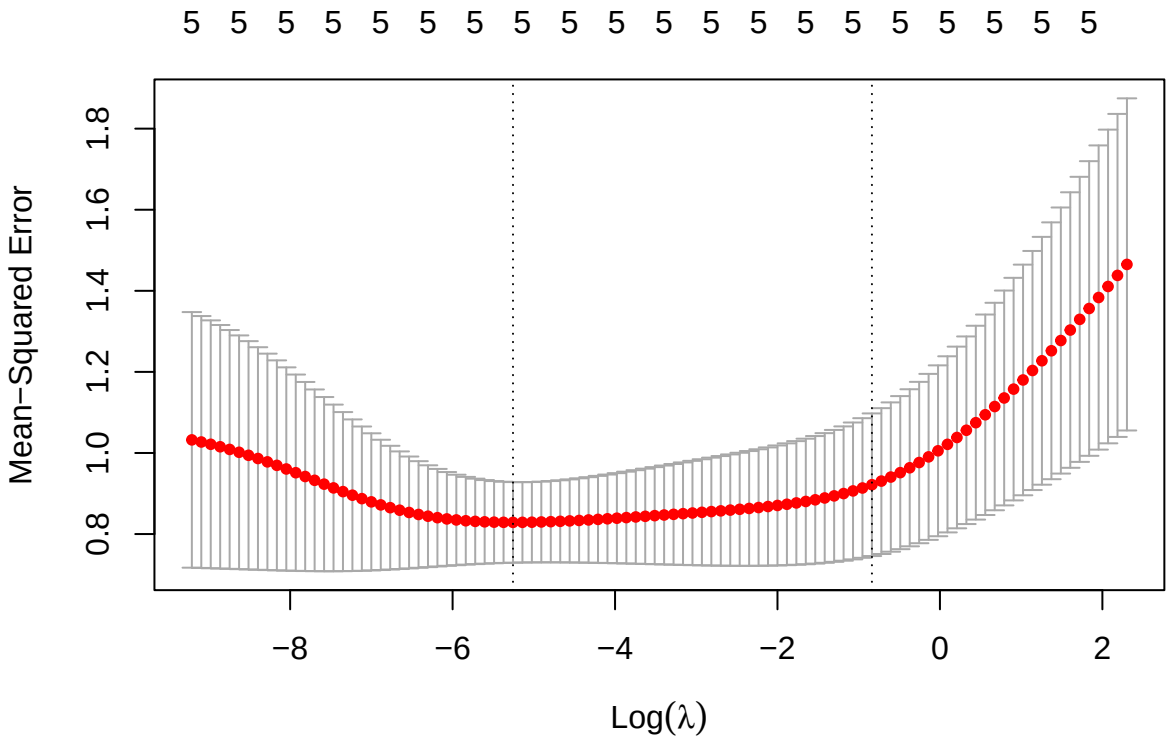
```
## Loaded glmnet 4.1-8
```

Ridge regression

We're going to do the ridge regression first, using a cross validation k-fold to choose the best value for λ.

```r
# Cross validation
lambdas_log <- 10^seq(-4, 1, length.out = 100)
cv_ridge <- cv.glmnet(X, y, alpha=0, lambda = (lambdas_log), standardize = TRUE)

plot(cv_ridge)
```



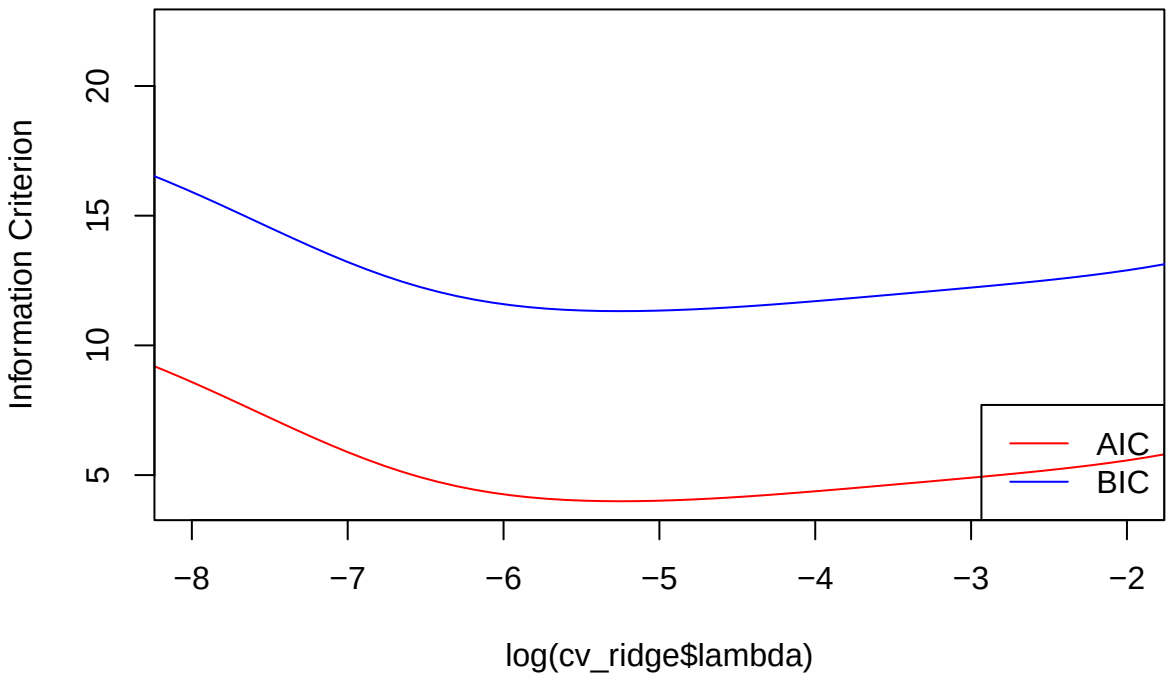

```
best_lambda <- cv_ridge$lambda.min # lambda that gives the lowest MSE
print(paste("The best value for lambda is", best_lambda))
```

```
## [1] "The best value for lambda is 0.00521400828799968"
```

Let's use AIC and BIC criteria to recheck this value.

```
# AIC and BIC
n <- nrow(X)
p <- ncol(X)
aic <- n * log(cv_ridge$cvm) + 2 * p
bic <- n * log(cv_ridge$cvm) + log(n) * p

plot(log(cv_ridge$lambda), aic, col = "red1", type = "l", xlim = c(-8, -2), ylab = "Information Criteri
lines(log(cv_ridge$lambda), bic, col = "blue1")
legend("bottomright", lwd = 1, col = c("red1", "blue1"), legend = c("AIC", "BIC"))
```




```
best_lambda_aic <- cv_ridge$lambda[which.min(aic)] # lambda that gives the lowest AIC
best_lambda_bic <- cv_ridge$lambda[which.min(bic)] # lambda that gives the lowest BIC
```

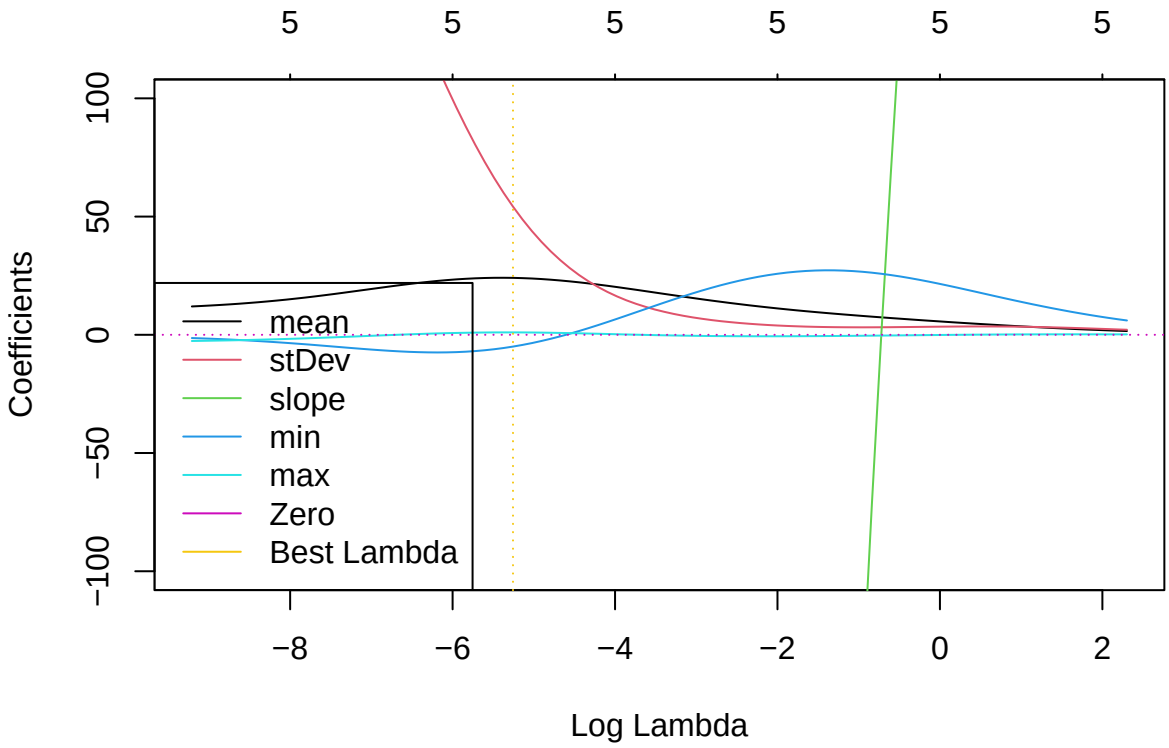
Now we can compare the different values for λ we found.

```
lambda_values <- c(best_lambda, best_lambda_aic, best_lambda_bic)
lambda_values
```

```
## [1] 0.005214008 0.005214008 0.005214008
```

There are the same. We can also plot the Regularization Path.

```
plot(cv_ridge$glmnet.fit, xvar = "lambda", ylim = c(-100, 100))
abline(h = 0, col = 6, lty = 3)
abline(v = log(best_lambda), col = 7, lty = 3)
legend("bottomleft", legend = c(colnames(X), "Zero", "Best Lambda"), col = 1:7, lty = 1)
```



Now we have the best value for λ, we do another ridge regression with this parameter and there is its results :

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept) -6.936572e-01
## mean         2.407840e+01
## stDev        5.438060e+01
## slope       -1.728896e+04
## min         -5.019195e+00
## max          1.013361e+00
```

```
predictions <- predict(best_model_ridge, newx = X)

# RMSE
rmse <- sqrt(mean((predictions - y)^2))
print(paste("RMSE ridge model :", rmse))
```

```
## [1] "RMSE ridge model : 0.763980768810407"
```

```r
# R^2
r_squared <- 1 - sum((y - predictions)^2) / sum((y - mean(y))^2)
print(paste("R^2 ridge model :", r_squared))
```

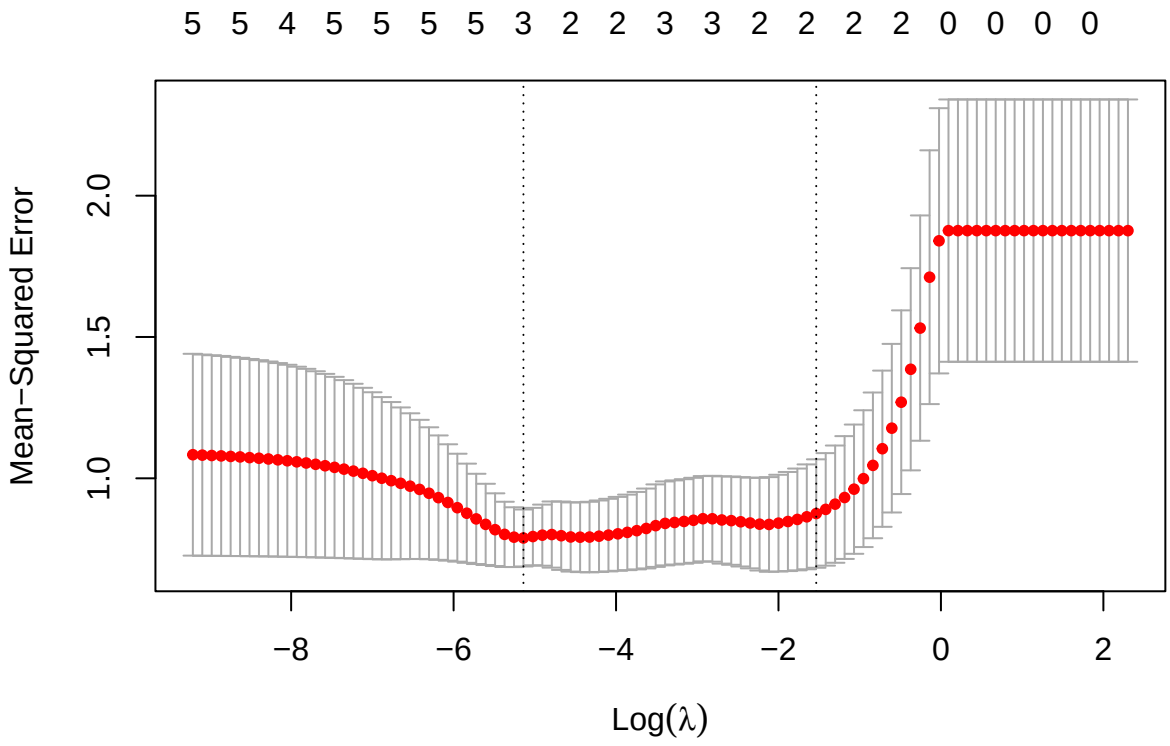
```
## [1] "R^2 ridge model : 0.668662382577303"
```

Conclusion of Ridge regression We can see that the coefficient of the slope is very important in absolute value relatively to the others (variables are scaled). It means that the slope is a very important feature to predict the fat value of a cookie. Futhermore, the coefficient of the mean, the standard deviation and the minimum are not null but negligible compared to the slope. It means that these features are not very important to predict the fat value of a cookie, but are more important than the max.

Lasso regression

```r
cv_lasso <- cv.glmnet(X, y, alpha=1, lambda = (lambdas_log), standardize = TRUE)

plot(cv_lasso)
```



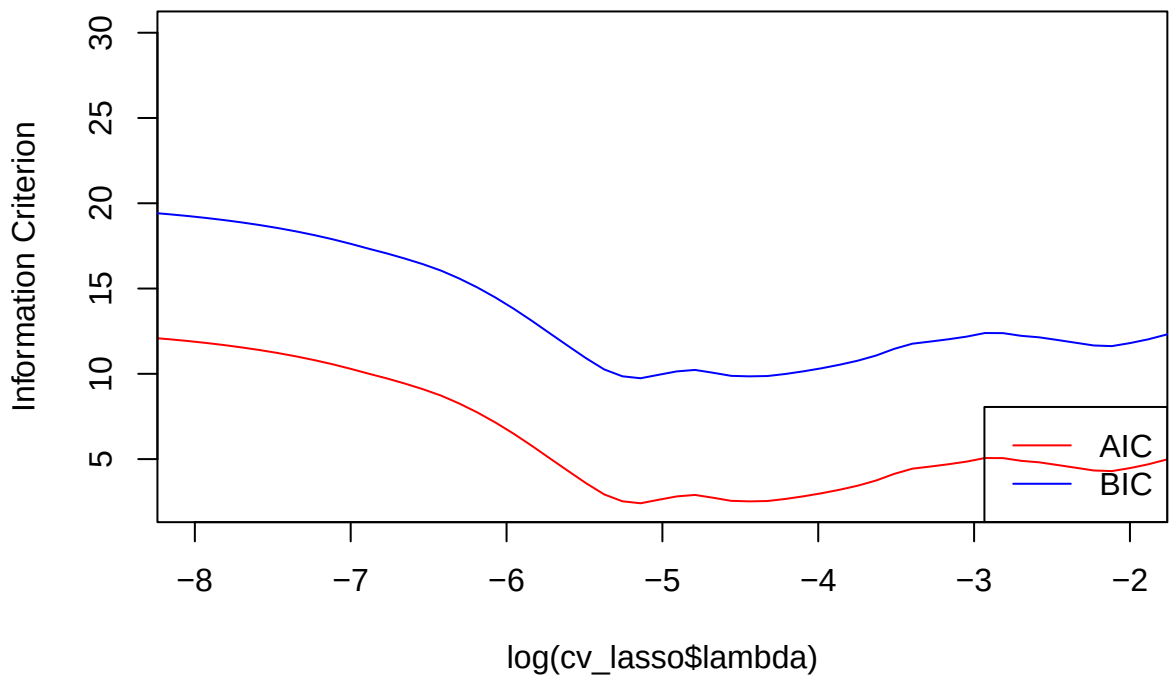
```r
best_lambda_lasso <- cv_lasso$lambda.min # lambda that gives the lowest MSE
print(paste("The best value for lambda is", best_lambda_lasso))
```

```
## [1] "The best value for lambda is 0.00585702081805667"
```

Let's use AIC and BIC criteria to recheck this value.

```r
# AIC and BIC
n <- nrow(X)
p <- ncol(X)
lasso_aic <- n * log(cv_lasso$cvm) + 2 * p
lasso_bic <- n * log(cv_lasso$cvm) + log(n) * p

plot(log(cv_lasso$lambda), lasso_aic, col = "red1", type = "l", xlim = c(-8, -2), ylab = "Information C
lines(log(cv_lasso$lambda), lasso_bic, col = "blue1")
legend("bottomright", lwd = 1, col = c("red1", "blue1"), legend = c("AIC", "BIC"))
```




```r
best_lambda_lasso_aic <- cv_lasso$lambda[which.min(lasso_aic)] # lambda that gives the lowest AIC
best_lambda_lasso_bic <- cv_lasso$lambda[which.min(lasso_bic)] # lambda that gives the lowest BIC
```

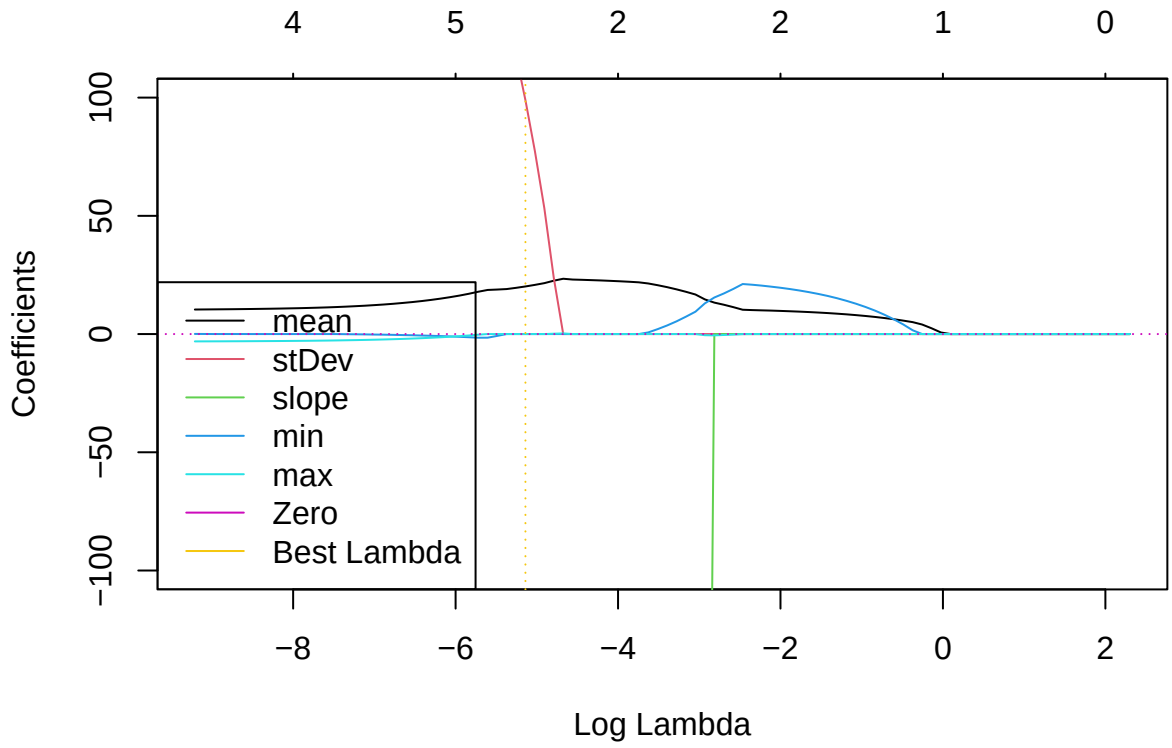
Now we can compare the different values for λ we found.

```r
lambda_lasso_values <- c(best_lambda_lasso, best_lambda_lasso_aic, best_lambda_lasso_bic)
lambda_lasso_values
```

```
## [1] 0.005857021 0.005857021 0.005857021
```

There are the same. We can also plot the Regularization Path.

```r
plot(cv_lasso$glmnet.fit, xvar = "lambda", ylim = c(-100, 100))
abline(h = 0, col = 6, lty = 3)
abline(v = log(best_lambda_lasso), col = 7, lty = 3)
legend("bottomleft", legend = c(colnames(X), "Zero", "Best Lambda"), col = 1:7, lty = 1)
```



Now we have the best value for λ, we do another lasso regression with this parameter and there is its results :

```r
best_model_lasso <- glmnet(X, y, alpha=1, lambda = best_lambda_lasso)
coef(best_model_lasso)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)     -1.445368
```

```
## mean            20.150945
## stDev           98.929333
## slope       -24211.100938
## min                 .
## max                 .
```

```r
predictions <- predict(best_model_lasso, newx = X)

# RMSE
rmse <- sqrt(mean((predictions - y)^2))
print(paste("RMSE lasso model :", rmse))
```

```
## [1] "RMSE lasso model : 0.7494688547651"
```

```r
# R^2
r_squared <- 1 - sum((y - predictions)^2) / sum((y - mean(y))^2)
print(paste("R^2 lasso model :", r_squared))
```

```
## [1] "R^2 lasso model : 0.681130433261201"
```

Conclusion of Lasso regression

We can see that the coefficient of the slope is also very important in absolute value relatively to the others (variables are scaled). It means that the slope is a very important feature to predict the fat value of a cookie. In this case, we can also see that the mean is not null wheras all the others are. It means that the mean could be a feature to predict the fat value of a cookie relatively to the others, but is less important than the slope.

No penalization

```r
model_linear <- lm(y ~ X)

predictions_linear <- predict(model_linear, newdata = data.frame(X))

# RMSE
rmse_linear <- sqrt(mean((predictions_linear - y)^2))
print(paste("RMSE linear model :", rmse_linear))
```

```
## [1] "RMSE linear model : 0.706440945059225"
```

```r
# R^2
r_squared_linear <- 1 - sum((y - predictions_linear)^2) / sum((y - mean(y))^2)
print(paste("R^2 linear model :", r_squared_linear))
```

```
## [1] "R^2 linear model : 0.716692796197633"
```

Conclusion

We can see that for both Lasso and Ridge regressions, the RMSE is higher and the R^2 is lower than those of the linear model. It means that the linear model is better than the Lasso and Ridge models to predict the fat value of a cookie in our study. There could be several reasons for that. First, the number of features is not very high, so the penalization is not very useful and we actually lost to much information. Secondly, the model was not overfitting. Still, the values of the RMSE and R^2 are very close, so the difference is not very important, and those penalizations show us that the slope and mean are indead the most important features to predict the fat value of a cookie.