Виконали: Кузьменко Юрій, Болотов Єгор

Лабораторна робота №10

Модельна діагностика.

Опис dataset

Назва dataset:

Spotify Top 10000 Streamed Songs

Link на dataset:

https://www.kaggle.com/datasets/rakkesharv/spotify-top-10000-streamed-songs

Опис dataset та постановку задачі:

Це набір даних, зібраний з веб-сайту Spotify, котрий містить потоки виконавця та кількість прослiховувань (було взято саме топ-10000) Основна мета: вплив факторів на популярність пісні й дізнатись найпопулярніших виконавців та треки.

Змінні та їх опис:

Position - Spotify Ranking

Artist Name - Artist Name

Song Name - Song Name

Days - No of days since the release of the song

Top 10 (xTimes) - No of times inside top 10

Peak Position - Peak position attained

Peak Position (xTimes) - No of times Peak position attained

Peak Streams - Total no of streams during Peak position

Total Streams - Total song streams

```
df <- read_csv("../Spotify_final_dataset.csv")
```

Завдання 1: Підготовка.

(A) Почистити dataset від всіх NA's за допомогою data <- na.omit(data)

```
df <- na.omit(df)
```

(B) Побудувати матрицю x;

```
x1 <- df$Peak_position_times
x2 <- df$Peak_streams
x3 <- df$Days
x4 <- df$Total_streams

x <- model.matrix(Peak_position ~ 0 + x1 + x2 + x3 + x4, data=df)
```

(C) Побудувати вектор відповідей y.

```
y <- df$Top_ten_times
```

Завдання 2: Побудова моделі Хребта.

(A) Побудувати модель регресії Хребта;

```
ridgeMod <- glmnet(x = x, y = y, alpha = 0)
ridgeMod
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 0)
##
##     Df  %Dev  Lambda
## 1    4  0.00 11510.0
## 2    4  0.45 10490.0
## 3    4  0.49  9556.0
## 4    4  0.54  8707.0
## 5    4  0.59  7934.0
## 6    4  0.65  7229.0
## 7    4  0.71  6587.0
## 8    4  0.78  6002.0
## 9    4  0.85  5468.0
## 10   4  0.94  4983.0
## 11   4  1.03  4540.0
## 12   4  1.12  4137.0
## 13   4  1.23  3769.0
## 14   4  1.35  3434.0
## 15   4  1.48  3129.0
## 16   4  1.62  2851.0
## 17   4  1.78  2598.0
## 18   4  1.95  2367.0
## 19   4  2.13  2157.0
## 20   4  2.33  1965.0
## 21   4  2.55  1791.0
## 22   4  2.79  1632.0
## 23   4  3.06  1487.0
## 24   4  3.34  1355.0
## 25   4  3.65  1234.0
## 26   4  3.99  1125.0
## 27   4  4.36  1025.0
## 28   4  4.76   933.6
## 29   4  5.20   850.7
## 30   4  5.67   775.1
## 31   4  6.18   706.3
## 32   4  6.73   643.5
## 33   4  7.33   586.4
## 34   4  7.97   534.3
## 35   4  8.67   486.8
## 36   4  9.42   443.6
## 37   4 10.22   404.2
## 38   4 11.08   368.2
## 39   4 12.00   335.5
## 40   4 12.98   305.7
```

```
## 41   4 14.03   278.6
## 42   4 15.14   253.8
## 43   4 16.31   231.3
## 44   4 17.56   210.7
## 45   4 18.86   192.0
## 46   4 20.23   174.9
## 47   4 21.67   159.4
## 48   4 23.16   145.2
## 49   4 24.71   132.3
## 50   4 26.30   120.6
## 51   4 27.95   109.9
## 52   4 29.63   100.1
## 53   4 31.35    91.2
## 54   4 33.09    83.1
## 55   4 34.84    75.7
## 56   4 36.61    69.0
## 57   4 38.36    62.9
## 58   4 40.11    57.3
## 59   4 41.84    52.2
## 60   4 43.53    47.6
## 61   4 45.19    43.3
## 62   4 46.80    39.5
## 63   4 48.35    36.0
## 64   4 49.85    32.8
## 65   4 51.28    29.9
## 66   4 52.64    27.2
## 67   4 53.93    24.8
## 68   4 55.15    22.6
## 69   4 56.29    20.6
## 70   4 57.36    18.8
## 71   4 58.36    17.1
## 72   4 59.28    15.6
## 73   4 60.14    14.2
## 74   4 60.93    12.9
## 75   4 61.67    11.8
## 76   4 62.34    10.7
## 77   4 62.96     9.8
## 78   4 63.53     8.9
## 79   4 64.06     8.1
## 80   4 64.54     7.4
## 81   4 64.98     6.7
## 82   4 65.39     6.1
## 83   4 65.77     5.6
## 84   4 66.13     5.1
## 85   4 66.46     4.6
## 86   4 66.76     4.2
## 87   4 67.05     3.9
## 88   4 67.33     3.5
## 89   4 67.59     3.2
## 90   4 67.84     2.9
## 91   4 68.08     2.7
## 92   4 68.31     2.4
```

```
## 93   4 68.54     2.2
## 94   4 68.76     2.0
## 95   4 68.97     1.8
## 96   4 69.17     1.7
## 97   4 69.37     1.5
## 98   4 69.57     1.4
## 99   4 69.76     1.3
## 100  4 69.94     1.2
```

(B) Зробити підсумок моделі за R^2, F-stat, RSE

```
mod <- lm(y~x)
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -155.588   -0.171   -0.039    0.073  140.211
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.317e-01  1.116e-01   2.973  0.00296 **
## xx1          1.548e+00  2.734e-02  56.638  < 2e-16 ***
## xx2         -1.176e-06  1.465e-07  -8.023 1.13e-15 ***
## xx3         -6.552e-02  1.837e-03 -35.666  < 2e-16 ***
## xx4          3.238e-07  5.167e-09  62.671  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.22 on 11075 degrees of freedom
## Multiple R-squared:  0.7227, Adjusted R-squared:  0.7226
## F-statistic:  7216 on 4 and 11075 DF,  p-value: < 2.2e-16
```

Дані результати стосуються регресійної моделі, яка має дуже високе значення F-статистики та дуже низький рівень значимості p-value, що означає, що є зв'язок між залежною та незалежними змінними в моделі (це все гіпотетично).

R^2 дорівнює 0.7227, що є не досить точно, але й має право на існування дана модель. RSE рівний 8.22

(C) Обчислити значення RSS

```
RSS <- sum(y-mod$fitted.values)^2
RSS
```

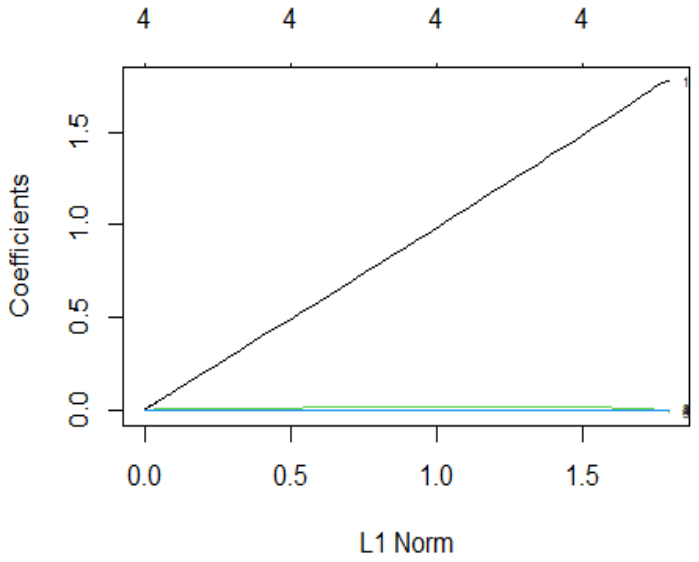
```
## [1] 1.663087e-22
```

Значення наближене до 0 каже нам про точність моделі

Завдання 3: Візуалізація моделі Хребта.

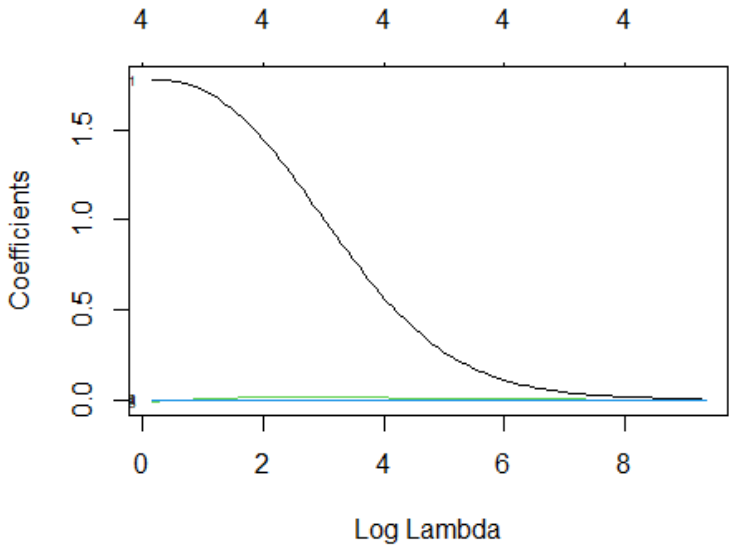
(A) Побудувати візуалізацію для xvar = “norm”;

```
plot(ridgeMod, xvar = "norm", label = TRUE)
```



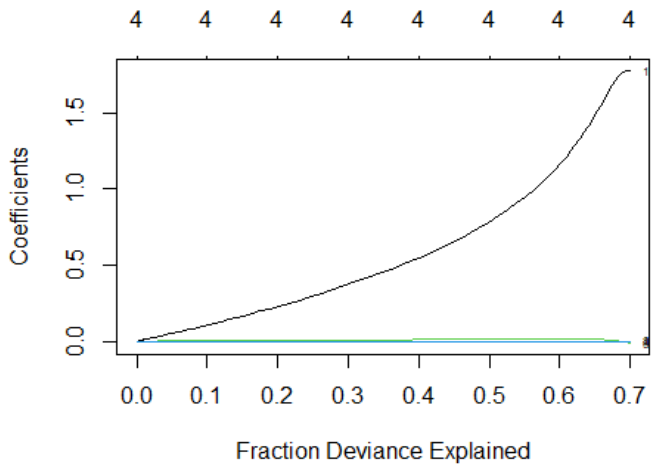
(B) Побудувати візуалізацію для xvar = ” lambda”;

```
plot(ridgeMod, label = TRUE, xvar = "lambda")
```



(C) Побудувати візуалізацію для xvar = “dev”;

```
plot(ridgeMod, label = TRUE, xvar = "dev")
```



(D) Визначити максимальне значення R^2;

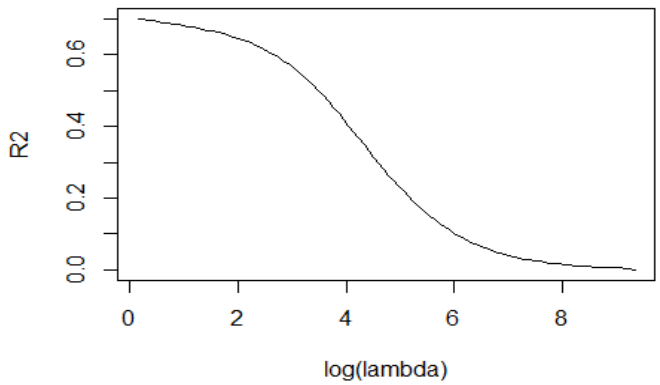
В районі 0.7

(E) Визначити три важливі фактори та вивести їх назви;

```
colnames(x)[c(1,2,4)]

## [1] "x1" "x2" "x4"
```

(F) Побудувати візуалізацію та виконати припуження про значення log(lambda) plot(log(ridgeMod$lambda), $ridgeMod$dev.ratio, type = “l”,xlab = “log(lambda)”, ylab = “R2”)

```
plot(log(ridgeMod$lambda), ridgeMod$dev.ratio, type = "l",
     xlab = "log(lambda)", ylab = "R2")
```



Завдання 4: Тонкі характеристики

(A) Визначити R^2 методом Хребта та порівняти з відповідним значенням в Lm() ridgeMod$dev.ratio[length(ridgeMod$dev.ratio)]

```
ridgeMod$dev.ratio[length(ridgeMod$dev.ratio)]
```

```
## [1] 0.6994264
```

Відрізняється від даного (0.7226)

(B) Визначити кількість вільних коефіцієнтів a0;

```
length(ridgeMod$a0)
```

```
## [1] 100
```

(C) Визначити кількість вільних коефіцієнтів beta;

```
length(ridgeMod$beta)
```

```
## [1] 400
```

(D) Визначити кількість коефіцієнтів lambda, які визначені автоматично;

```
length(ridgeMod$lambda)
```

```
## [1] 100
```

(E) Вивести значення коефіцієнтів для певного lambda;

```
coef(ridgeMod)[, 19]
```

```
##  (Intercept)           x1           x2           x3           x4
## 2.613982e+00 2.172946e-02 6.792044e-08 4.845529e-04 1.567944e-09
```

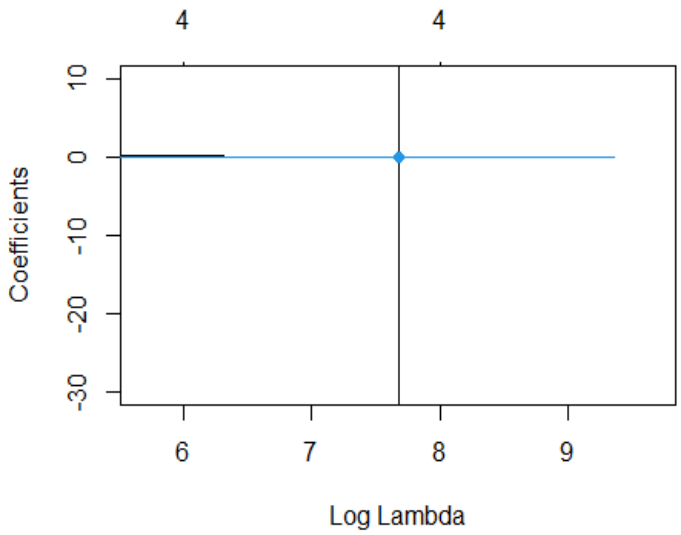
(F) Вивести значення lambda;

```
ridgeMod$lambda[19]
```

```
## [1] 2156.843
```

(G) Візуально представити коефіцієнти для відповідного lambda.

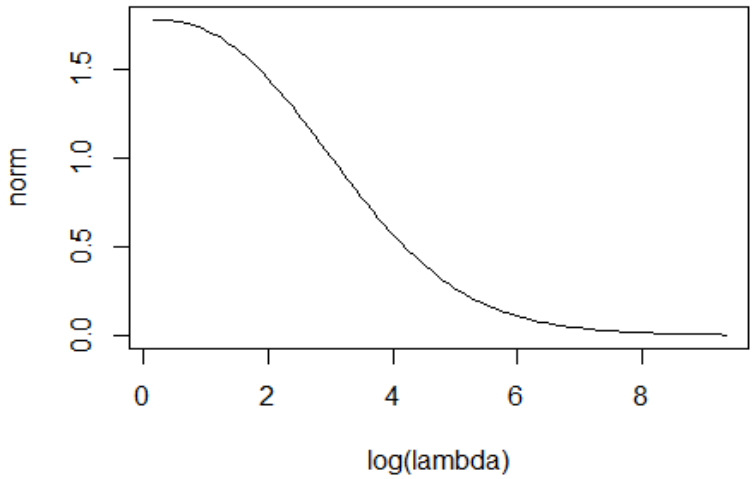
```
plot(ridgeMod, label = TRUE, xvar = "lambda",
     xlim = log(ridgeMod$lambda[19]) + c(-2, 2), ylim = c(-30, 10))
abline(v = log(ridgeMod$lambda[19]))
points(rep(log(ridgeMod$lambda[19]), nrow(ridgeMod$beta)), ridgeMod$beta[, 19],
       pch = 16, col = 1:6)
```




```
ridgeMod$beta[, 19]

##           x1           x2           x3           x4
## 2.172946e-02 6.792044e-08 4.845529e-04 1.567944e-09
```

(H) Візуальне представлення для l^2 норми
plot(log(ridgeMod$lambda), sqrt(colSums(ridgeMod$beta^2)), type = “l”, xlab = “log(lambda)”, ylab = “l2 norm”)

```
plot(log(ridgeMod$lambda), sqrt(colSums(ridgeMod$beta^2)), type = "l", xlab =
"log(lambda)", ylab = "l2
norm")
```



Завдання 5: Автоматичне налаштування

(A) Визначити мінімальне значення для lambda;

```
set.seed(7777)
kcvRidge <- cv.glmnet(x=x, y=y, alpha=0, nfolds = 10)
kcvRidge$lambda.min
```

```
## [1] 1.151042
```

(B) Визначити індекс для мінімального значення lambda;

```
indMinLambda <- which.min(kcvRidge$cvm)
kcvRidge$lambda[indMinLambda]
```

```
## [1] 1.151042
```

(C) Визначити значення мінімальної похибку CV;

```
ridgeMod$lambda[indMinLambda]
```

```
## [1] 1.151042
```

(D) Згенерувати сітку з часових інтервалів;

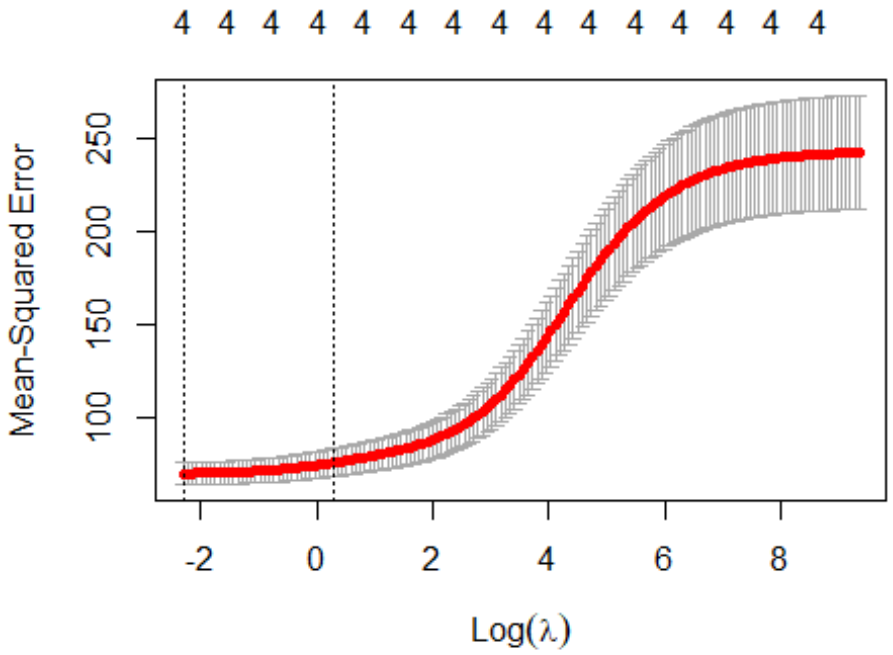
```
lambdaGrid <- 10^seq(log10(kcvRidge$lambda[1]), log10(0.1),
                     length.out = 150)
lambdaGrid
```

```
##   [1] 1.151042e+04 1.064447e+04 9.843673e+03 9.103117e+03 8.418274e+03
##   [6] 7.784953e+03 7.199278e+03 6.657664e+03 6.156797e+03 5.693611e+03
##  [11] 5.265271e+03 4.869156e+03 4.502841e+03 4.164085e+03 3.850814e+03
##  [16] 3.561111e+03 3.293202e+03 3.045449e+03 2.816335e+03 2.604457e+03
##  [21] 2.408520e+03 2.227323e+03 2.059757e+03 1.904798e+03 1.761497e+03
##  [26] 1.628977e+03 1.506426e+03 1.393095e+03 1.288290e+03 1.191370e+03
##  [31] 1.101741e+03 1.018855e+03 9.422052e+02 8.713216e+02 8.057706e+02
##  [36] 7.451511e+02 6.890921e+02 6.372506e+02 5.893091e+02 5.449744e+02
##  [41] 5.039751e+02 4.660602e+02 4.309977e+02 3.985730e+02 3.685877e+02
##  [46] 3.408582e+02 3.152149e+02 2.915008e+02 2.695707e+02 2.492904e+02
##  [51] 2.305359e+02 2.131923e+02 1.971535e+02 1.823213e+02 1.686049e+02
##  [56] 1.559205e+02 1.441903e+02 1.333427e+02 1.233111e+02 1.140342e+02
##  [61] 1.054552e+02 9.752162e+01 9.018490e+01 8.340014e+01 7.712581e+01
##  [66] 7.132350e+01 6.595772e+01 6.099561e+01 5.640681e+01 5.216323e+01
##  [71] 4.823890e+01 4.460981e+01 4.125374e+01 3.815015e+01 3.528005e+01
##  [76] 3.262587e+01 3.017137e+01 2.790153e+01 2.580245e+01 2.386129e+01
##  [81] 2.206616e+01 2.040609e+01 1.887091e+01 1.745122e+01 1.613833e+01
##  [86] 1.492422e+01 1.380144e+01 1.276314e+01 1.180295e+01 1.091499e+01
##  [91] 1.009384e+01 9.334461e+00 8.632213e+00 7.982798e+00 7.382238e+00
##  [96] 6.826860e+00 6.313264e+00 5.838307e+00 5.399081e+00 4.992899e+00
## [101] 4.617275e+00 4.269910e+00 3.948677e+00 3.651611e+00 3.376895e+00
## [106] 3.122845e+00 2.887908e+00 2.670646e+00 2.469729e+00 2.283927e+00
```

```
## [111] 2.112103e+00 1.953206e+00 1.806263e+00 1.670375e+00 1.544710e+00
## [116] 1.428499e+00 1.321030e+00 1.221647e+00 1.129741e+00 1.044748e+00
## [121] 9.661501e-01 8.934650e-01 8.262482e-01 7.640881e-01 7.066045e-01
## [126] 6.534454e-01 6.042856e-01 5.588242e-01 5.167829e-01 4.779045e-01
## [131] 4.419509e-01 4.087022e-01 3.779549e-01 3.495207e-01 3.232257e-01
## [136] 2.989089e-01 2.764214e-01 2.556258e-01 2.363946e-01 2.186103e-01
## [141] 2.021639e-01 1.869547e-01 1.728898e-01 1.598830e-01 1.478547e-01
## [146] 1.367314e-01 1.264449e-01 1.169322e-01 1.081352e-01 1.000000e-01
```

(E) Побудувати візуазізацію для $log\ \lambda$

```
kcvRidge2 <- cv.glmnet(x = x, y = y, nfolds = 10, alpha = 0,
                       lambda = lambdaGrid)
plot(kcvRidge2)
```



(F) Визначити мінімальне значення CV

```
kcvRidge2$lambda.min
```

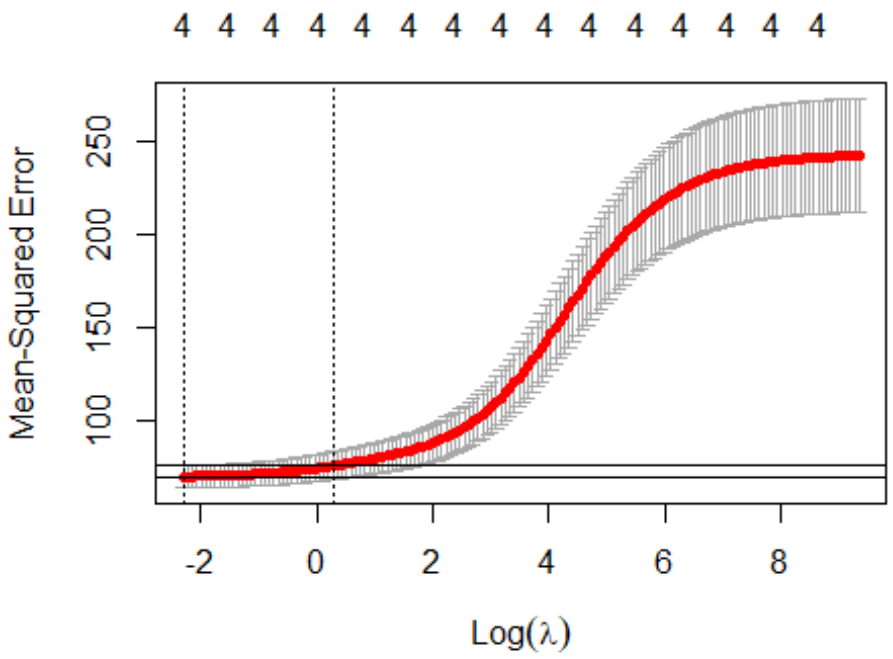
```
## [1] 0.1
```

```
kcvRidge2$lambda.1se
```

```
## [1] 1.32103
```

(G) Візуалізувати найнижче значення CV та його стандартне відхилення;

```
plot(kcvRidge2)
indMin2 <- which.min(kcvRidge2$cvm)
abline(h = kcvRidge2$cvm[indMin2] + c(0, kcvRidge2$cvsd[indMin2]))
```

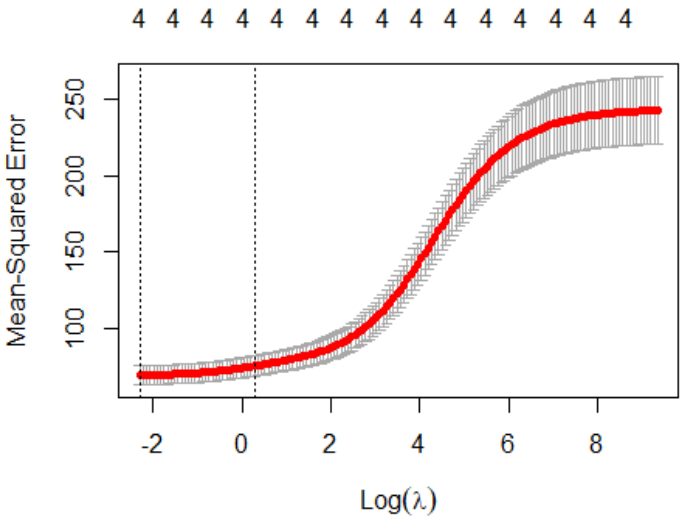


(H) Визначити оптимальний CV для найбільшої кількості груп nfolds = nrow(data)

```
ncvRidge <- cv.glmnet(x = x, y = y, alpha = 0, nfolds = nrow(df),
                      lambda = lambdaGrid)

## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations
per
## fold

ncvRidge$lambda.min

## [1] 0.1
```

(I) Виконати візуалізацію для (H)

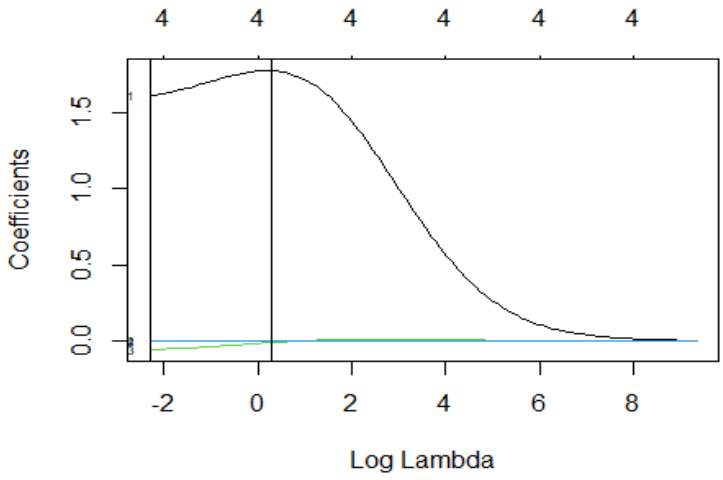
```
plot(ncvRidge)
```



Завдання 6: Прогнозування

(A) Визначити значення коефіцієнтів моделі за CV для відповідного lambda;

```
modRidgeCV <- kcvRidge2$glmnet.fit
plot(modRidgeCV, label = TRUE, xvar = "lambda")
abline(v = log(c(kcvRidge2$lambda.min, kcvRidge2$lambda.1se)))
```



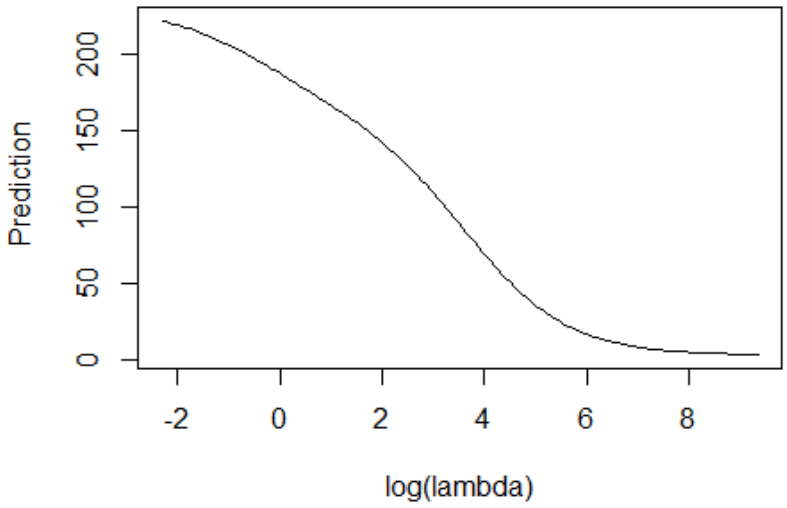

```
predict(modRidgeCV, type = "coefficients", s = kcvRidge2$lambda.1se)

## 5 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept) -8.554195e-01
## x1           1.779118e+00
## x2           7.484359e-07
## x3          -9.578479e-03
## x4           1.626421e-07
```

(B) Виконати прогнох для перших двої значень;

```
predict(modRidgeCV, type = "response", s = kcvRidge2$lambda.1se,
        newx = x[1:2, ])

##         s1
## 1 181.5723
## 2 160.9527
```

(C) Звізуалізувати на скільки зміняться значення прогнозу для різних значень lambda.

```
plot(log(modRidgeCV$lambda),
     predict(modRidgeCV, type = "response", newx = x[1, , drop = FALSE]),
     type = "l", xlab = "log(lambda)", ylab = " Prediction")
```


Prediction
0
50
100
150
200
-2
0
2
4
6
8
log(lambda)