

```

# Load required Libraries
library("gclus") # For hierarchical clustering

## Loading required package: cluster

library("glmnet") # For LASSO regression

## Loading required package: Matrix

## Loaded glmnet 4.1-8

library("RcmdrMisc")

## Loading required package: car

## Loading required package: carData

## Loading required package: sandwich

# Load the body dataset if available
data(body)

# View the first few rows of the 'body' dataset
head(body)

##   Biacrom Biiliac Bitro ChestDp ChestD ElbowD WristD KneeD AnkleD
ShoulderG
## 1    42.9    26.0  31.5    17.7   28.0   13.1   10.4   18.8   14.1
106.2
## 2    43.7    28.5  33.5    16.9   30.8   14.0   11.8   20.6   15.1
110.5
## 3    40.1    28.2  33.3    20.9   31.7   13.9   10.9   19.7   14.1
115.1
## 4    44.3    29.9  34.0    18.4   28.2   13.9   11.2   20.9   15.0
104.5
## 5    42.5    29.9  34.0    21.5   29.4   15.2   11.6   20.7   14.9
107.5
## 6    43.3    27.0  31.5    19.6   31.3   14.0   11.5   18.8   13.9
119.8
##   ChestG WaistG AbdG HipG ThighG BicepG ForearmG KneeG CalfG AnkleG WristG
Age
## 1    89.5    71.5  74.5  93.5    51.5    32.5      26.0   34.5   36.5   23.5   16.5
21
## 2    97.0    79.0  86.5  94.8    51.5    34.4      28.0   36.5   37.5   24.5   17.0
23
## 3    97.5    83.2  82.9  95.0    57.3    33.4      28.8   37.0   37.3   21.9   16.9
28
## 4    97.0    77.8  78.8  94.0    53.0    31.0      26.2   37.0   34.8   23.0   16.6
23
## 5    97.5    80.0  82.5  98.5    55.4    32.0      28.4   37.7   38.6   24.4   18.0
22

```

```

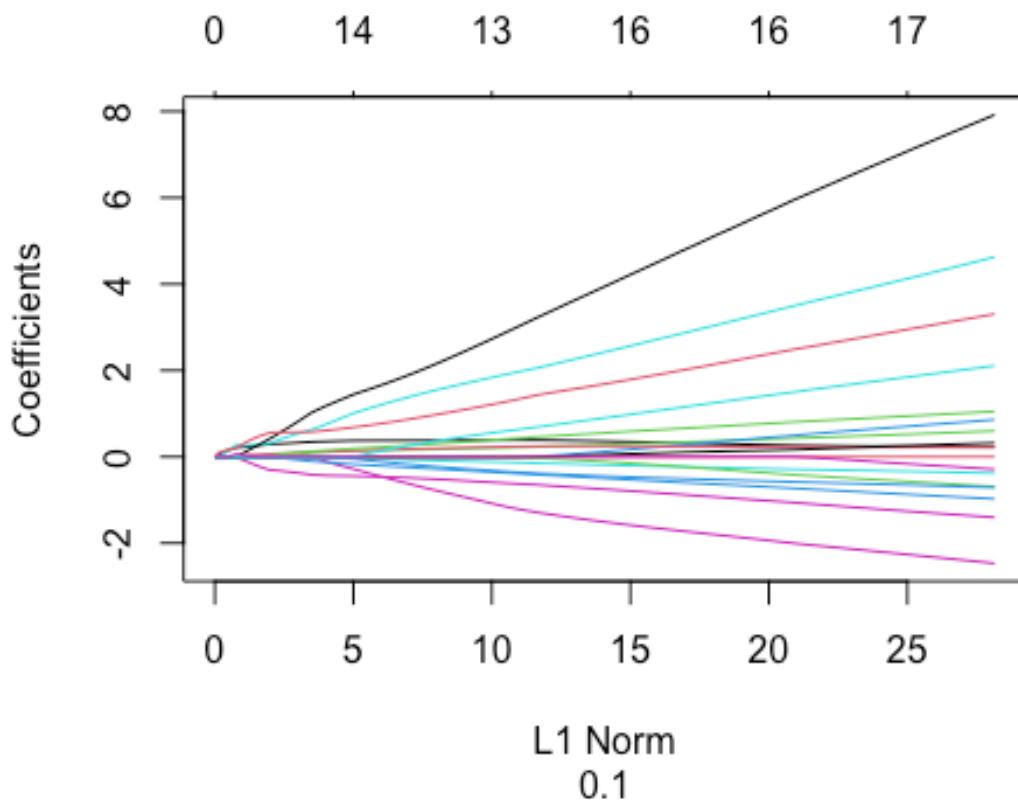
## 6   99.9   82.5  80.1  95.3   57.5   33.0    28.0   36.6   36.1   23.5   16.9
21
##   Weight Height Gender
## 1   65.6   174.0      1
## 2   71.8   175.3      1
## 3   80.7   193.5      1
## 4   72.6   186.5      1
## 5   78.8   187.2      1
## 6   74.8   181.5      1

# Convert predictors and outcome variable to matrix format
X_Matrix <- as.matrix(body[, 1:21]) # Extract predictor variables
yResp <- as.matrix(body[,25]) # Extract outcome variable

# Fit Logistic regression model with LASSO regularization
GLM_Net1 <- glmnet(
  X_Matrix,
  yResp,
  family = "binomial"
)

# Plot coefficients of the fitted model
plot(GLM_Net1, s = 0.1)

```



```

# View coefficients at a specific value of the regularization parameter (s)
coef(GLM_Net1, s = 0.1)

## 22 x 1 sparse Matrix of class "dgCMatrix"
##                                     s1
## (Intercept) -19.777687812
## Biacrom      0.210662887
## Biiliac     .
## Bitro        .
## ChestDp     .
## ChestD      .
## ElbowD      0.269346765
## WristD      .
## KneeD       .
## AnkleD      .
## ShoulderG   0.019843915
## ChestG      .
## WaistG      .
## AbdG        .
## HipG        .
## ThighG      -0.009236944
## BicepG      .
## ForearmG    0.207735326
## KneeG       .
## CalfG        .
## AnkleG      .
## WristG      0.057088005

# Perform cross-validated LASSO regression
GLM_Net_CV <- cv.glmnet(X_Matrix, yResp, family = "binomial")

# View coefficients from cross-validated model at a specific value of s
coef(GLM_Net_CV, s = 0.1)

## 22 x 1 sparse Matrix of class "dgCMatrix"
##                                     s1
## (Intercept) -19.777687812
## Biacrom      0.210662887
## Biiliac     .
## Bitro        .
## ChestDp     .
## ChestD      .
## ElbowD      0.269346765
## WristD      .
## KneeD       .
## AnkleD      .
## ShoulderG   0.019843915
## ChestG      .
## WaistG      .
## AbdG        .

```

```

## HipG      .
## ThighG    -0.009236944
## BicepG    .
## ForearmG  0.207735326
## KneeG     .
## CalfG     .
## AnkleG    .
## WristG    0.057088005

# Predict probabilities using cross-validated model
Predprobs <- predict(GLM_Net_CV, newx = X_Matrix, s = 0.1, type = "response")

# Convert predicted probabilities to predicted classes using different
# thresholds
PredGender <- 1 * (Predprobs > 0.5) # Threshold: 0.5
table(yResp, PredGender) # Compare predicted vs. actual classes

##      PredGender
## yResp  0  1
##       0 243 17
##       1 19 228

PredGender1 <- 1 * (Predprobs > 0.75) # Threshold: 0.75
table(yResp, PredGender1) # Compare predicted vs. actual classes

##      PredGender1
## yResp  0  1
##       0 257  3
##       1 79 168

PredGender2 <- 1 * (Predprobs > 0.9) # Threshold: 0.9
table(yResp, PredGender2) # Compare predicted vs. actual classes

##      PredGender2
## yResp  0  1
##       0 260  0
##       1 199 48

PredGender3 <- 1 * (Predprobs > 0.35) # Threshold: 0.35
table(yResp, PredGender3) # Compare predicted vs. actual classes

##      PredGender3
## yResp  0  1
##       0 223 37
##       1  3 244

PredGender4 <- 1 * (Predprobs > 0.11) # Threshold: 0.11
table(yResp, PredGender4) # Compare predicted vs. actual classes

##      PredGender4
## yResp  0  1

```

```

##      0 64 196
##      1 0 247

##### thursday's class

# Load the mtcars dataset from the datasets package and convert the 'am'
variable to a factor
data(mtcars, package="datasets")
mtcars <- within(mtcars, {
  am <- as.factor(am)
})

# Fit a Logistic regression model predicting 'am' (automatic/manual
transmission) based on 'disp' (displacement), 'hp' (horsepower), 'mpg' (miles
per gallon), and 'wt' (weight)
lmof_am <- glm(am ~ disp + hp + mpg + wt, family=binomial(logit),
data=mtcars)
summary(lmof_am) # Display summary of the logistic regression model

##
## Call:
## glm(formula = am ~ disp + hp + mpg + wt, family = binomial(logit),
##      data = mtcars)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -18.48207   40.90451  -0.452   0.651
## disp        -0.02588    0.04087  -0.633   0.527
## hp          0.10871    0.09837   1.105   0.269
## mpg         1.13503    1.55720   0.729   0.466
## wt          -4.80560   3.97978  -1.208   0.227
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 43.230 on 31 degrees of freedom
## Residual deviance: 8.162 on 27 degrees of freedom
## AIC: 18.162
##
## Number of Fisher Scoring iterations: 9

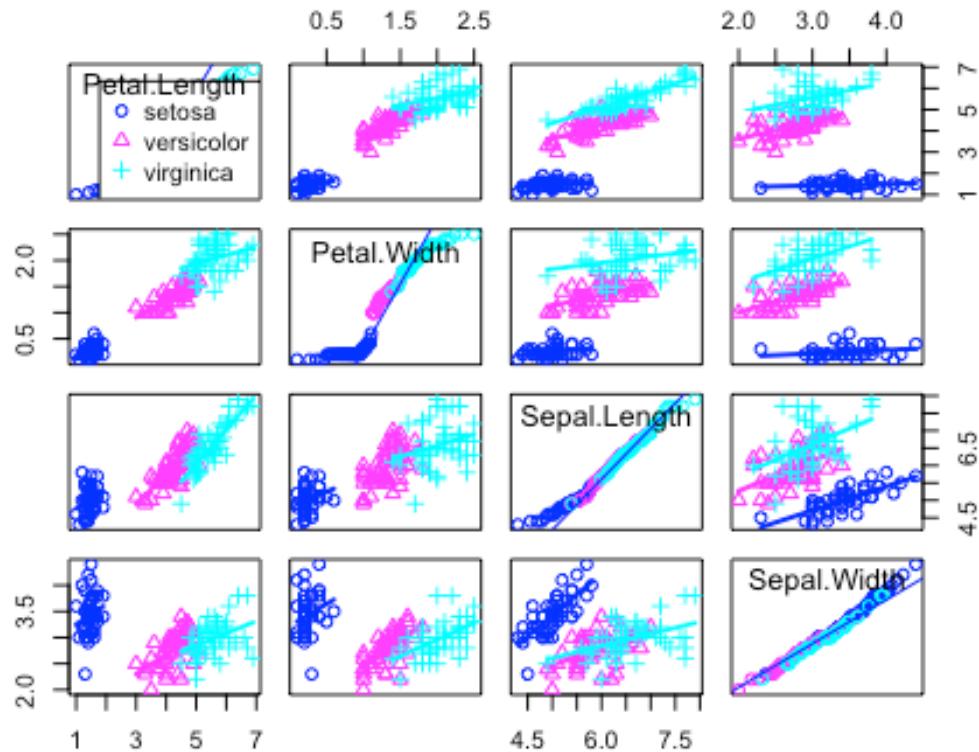
exp(coef(lmof_am)) # Exponentiated coefficients ("odds ratios")

## (Intercept)           disp           hp           mpg           wt
## 9.404558e-09 9.744539e-01 1.114835e+00 3.111263e+00 8.183786e-03

# Load the iris dataset from the datasets package and create a scatterplot
matrix with separate panels for each species
data(iris, package="datasets")
scatterplotMatrix(~Petal.Length+Petal.Width+Sepal.Length+Sepal.Width |
Species,
  regLine=TRUE, smooth=FALSE,

```

```
diagonal=list(method="qqplot"), by.groups=TRUE,
  data=iris)
```



```
# Fit a Logistic regression model predicting 'Species' based on
# 'Petal.Length', 'Petal.Width', 'Sepal.Length', and 'Sepal.Width'
lm_species <- glm(Species ~ Petal.Length + Petal.Width + Sepal.Length +
Sepal.Width,
  family=binomial(logit), data=iris)

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(lm_species) # Display summary of the Logistic regression model

##
## Call:
## glm(formula = Species ~ Petal.Length + Petal.Width + Sepal.Length +
##     Sepal.Width, family = binomial(logit), data = iris)
## 

## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 16.946 457457.097      0      1
## Petal.Length 20.088 107724.589      0      1
```

```

## Petal.Width      21.608 154350.604      0      1
## Sepal.Length    -11.759 130504.037      0      1
## Sepal.Width     -7.842  59415.373      0      1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1.9095e+02  on 149  degrees of freedom
## Residual deviance: 3.2940e-09  on 145  degrees of freedom
## AIC: 10
##
## Number of Fisher Scoring iterations: 25

exp(coef(lm_species)) # Exponentiated coefficients ("odds ratios")

## (Intercept) Petal.Length Petal.Width Sepal.Length Sepal.Width
## 2.287827e+07 5.297991e+08 2.421488e+09 7.816945e-06 3.930486e-04

# Add fitted values from the Logistic regression model to the iris dataset
iris <- within(iris, {
  fitted.lm_species <- fitted(lm_species)
})

# Fit a multinomial Logistic regression model predicting 'Species' based on
# 'Petal.Length', 'Petal.Width', 'Sepal.Length', and 'Sepal.Width'
library(nnet, pos=16)
MLM_on_species <- multinom(Species ~ Petal.Length + Petal.Width +
Sepal.Length + Sepal.Width, data=iris,
  trace=FALSE)
summary(MLM_on_species, cor=FALSE, Wald=TRUE) # Display summary of the
multinomial Logistic regression model

## Call:
## multinom(formula = Species ~ Petal.Length + Petal.Width + Sepal.Length +
##           Sepal.Width, data = iris, trace = FALSE)
##
## Coefficients:
##             (Intercept) Petal.Length Petal.Width Sepal.Length Sepal.Width
## versicolor    18.69037     14.24477   -3.097684   -5.458424   -8.707401
## virginica    -23.83628     23.65978   15.135301   -7.923634  -15.370769
##
## Std. Errors:
##             (Intercept) Petal.Length Petal.Width Sepal.Length Sepal.Width
## versicolor    34.97116     60.19170   45.48852    89.89215   157.0415
## virginica     35.76649     60.46753   45.93406    89.91153   157.1196
##
## Value/SE (Wald statistics):
##             (Intercept) Petal.Length Petal.Width Sepal.Length Sepal.Width
## versicolor    0.5344511     0.2366567  -0.06809815  -0.06072192  -0.05544649
## virginica    -0.6664417     0.3912807   0.32950063  -0.08812701  -0.09782845
##

```

```

## Residual Deviance: 11.89973
## AIC: 31.89973

# Add fitted values from the multinomial Logistic regression model to the
# iris dataset
iris <- within(iris, {
  fitted.MLM_on_species <- fitted(MLM_on_species)
})

head(iris) # Display the first few rows of the modified iris dataset

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
fitted.lm_species
## 1          5.1       3.5        1.4       0.2  setosa
2.220446e-16
## 2          4.9       3.0        1.4       0.2  setosa
2.220446e-16
## 3          4.7       3.2        1.3       0.2  setosa
2.220446e-16
## 4          4.6       3.1        1.5       0.2  setosa
1.875686e-12
## 5          5.0       3.6        1.4       0.2  setosa
2.220446e-16
## 6          5.4       3.9        1.7       0.4  setosa
2.220446e-16
##   fitted.MLM_on_species.setosa fitted.MLM_on_species.versicolor
## 1                  1.000000e+00                  1.526406e-09
## 2                  9.999996e-01                  3.536476e-07
## 3                  1.000000e+00                  4.443506e-08
## 4                  9.999968e-01                  3.163905e-06
## 5                  1.000000e+00                  1.102983e-09
## 6                  1.000000e+00                  3.521573e-10
##   fitted.MLM_on_species.virginica
## 1                  2.716417e-36
## 2                  2.883729e-32
## 3                  6.103424e-34
## 4                  7.117010e-31
## 5                  1.289946e-36
## 6                  1.344907e-35

```