# logistic

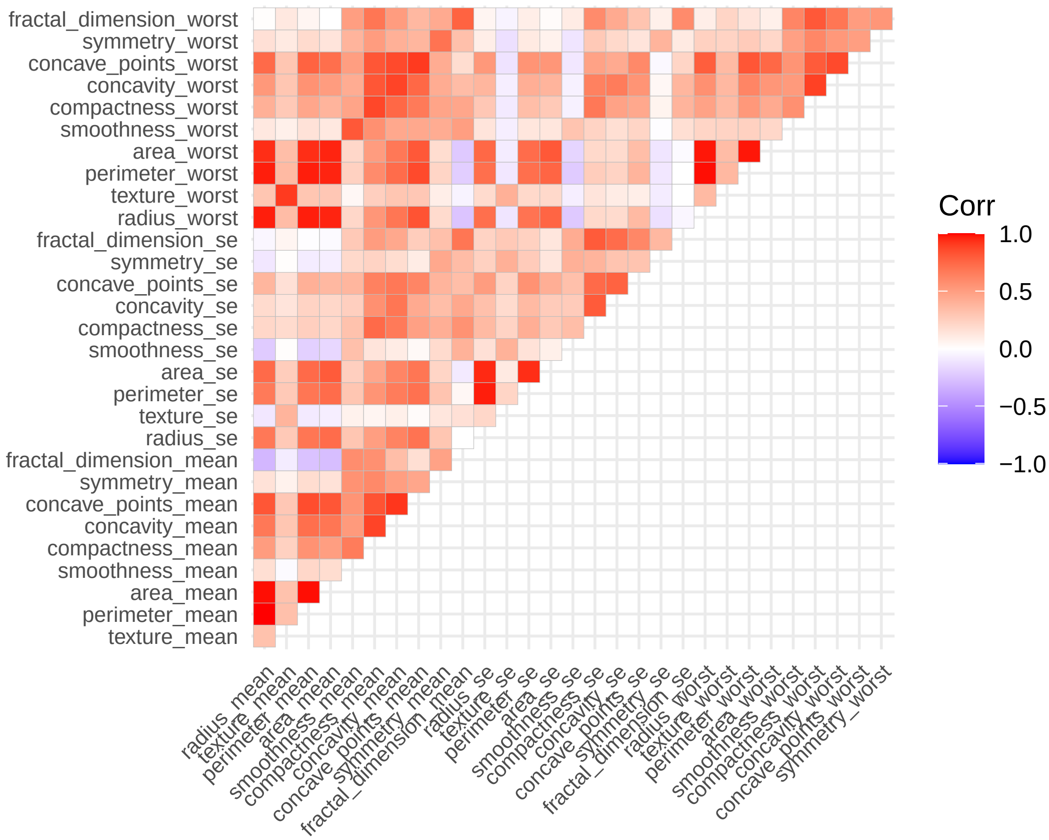
Xinran Sun, Haotian Wu, Lin Yang, Shengzhi Luo

3/17/2022

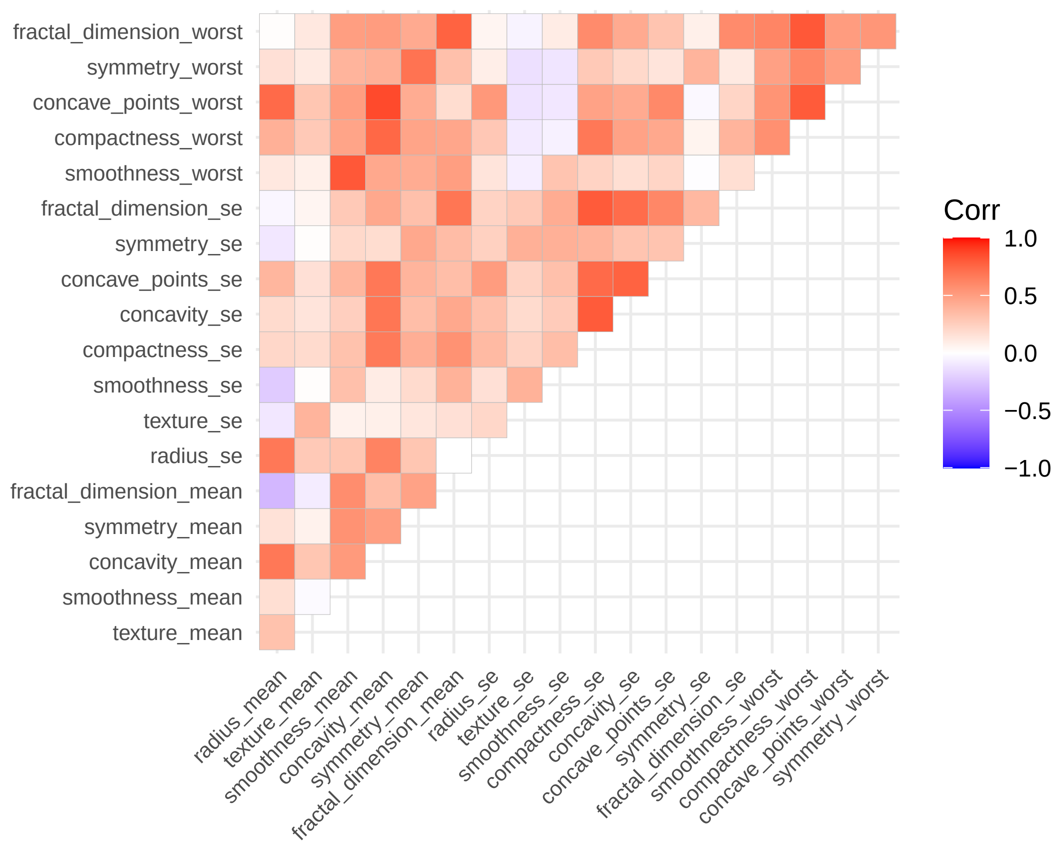
```
ggplot2::theme_set(theme_minimal() + theme(legend.position = "bottom"))
```

## data import and data clean

```
#load the data
breast = read.csv("breast-cancer.csv") %>%
  janitor::clean_names() %>%
  dplyr::select(-1, -33) %>% #drop id and NA columns
  mutate(diagnosis = recode(diagnosis, "M" = 1, "B" = 0))

#check collinearity
corr = breast[2:31] %>%
  cor()

ggcorrplot(corr, type = "upper", tl.cex = 8)
```




```
#remove some highly correlated variables
breast_dat <- breast %>% dplyr::select(-area_se, -perimeter_se, -area_worst, -perimeter_mean, -perimete

corr1 = breast_dat[2:20] %>%
  cor()

ggcorrplot(corr1, type = "upper", tl.cex = 8)
```




```r
#partition data into training and test data
trainRows <- createDataPartition(y = breast_dat$diagnosis, p = 0.8, list = FALSE)
breast_train <- breast_dat[trainRows, ]
breast_test <-  breast_dat[-trainRows, ]

head(breast_dat, 5)
```

```
##   diagnosis radius_mean texture_mean smoothness_mean concavity_mean
## 1         1       17.99        10.38         0.11840         0.3001
## 2         1       20.57        17.77         0.08474         0.0869
## 3         1       19.69        21.25         0.10960         0.1974
## 4         1       11.42        20.38         0.14250         0.2414
## 5         1       20.29        14.34         0.10030         0.1980
##   symmetry_mean fractal_dimension_mean radius_se texture_se smoothness_se
## 1        0.2419                0.07871    1.0950     0.9053      0.006399
## 2        0.1812                0.05667    0.5435     0.7339      0.005225
## 3        0.2069                0.05999    0.7456     0.7869      0.006150
## 4        0.2597                0.09744    0.4956     1.1560      0.009110
## 5        0.1809                0.05883    0.7572     0.7813      0.011490
##   compactness_se concavity_se concave_points_se symmetry_se
## 1        0.04904      0.05373           0.01587     0.03003
## 2        0.01308      0.01860           0.01340     0.01389
## 3        0.04006      0.03832           0.02058     0.02250
## 4        0.07458      0.05661           0.01867     0.05963
## 5        0.02461      0.05688           0.01885     0.01756
##   fractal_dimension_se smoothness_worst compactness_worst concave_points_worst
```

```
## 1              0.006193           0.1622           0.6656               0.2654
## 2              0.003532           0.1238           0.1866               0.1860
## 3              0.004571           0.1444           0.4245               0.2430
## 4              0.009208           0.2098           0.8663               0.2575
## 5              0.005115           0.1374           0.2050               0.1625
##   symmetry_worst fractal_dimension_worst
## 1         0.4601                 0.11890
## 2         0.2750                 0.08902
## 3         0.3613                 0.08758
## 4         0.6638                 0.17300
## 5         0.2364                 0.07678
```

```
r = dim(breast_dat)[1] #row number
c = dim(breast_dat)[2] #column number

var_names = names(breast_dat)[-c(1,2)] #variable names

standardize = function(col) {
  mean = mean(col)
  sd = sd(col)
  return((col - mean)/sd)
}

stand_df = breast_dat %>%
  dplyr::select(radius_mean:fractal_dimension_worst) %>%
  map_df(.x = ., standardize) #standardize

X = stand_df #predictors
y = as.vector(ifelse(breast_dat[,2] == "M", 1, 0))#response
```

```
x_train <- breast_train[2:20] #predictors
y_train <- breast_train[1] #response
x_train_stan <- cbind(rep(1, nrow(x_train)), scale(x_train))

x_test <- breast_test[2:20]
x_test_stan <- cbind(rep(1, nrow(x_test)), scale(x_test))
```

### Full logistic model

```
glm.fit <- glm(diagnosis ~ .,
               data = breast_dat,
               subset = trainRows,
               family = binomial(link = "logit"))
summary(glm.fit)
```

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial(link = "logit"),
##     data = breast_dat, subset = trainRows)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -1.43985  -0.02891  -0.00199   0.00015   2.86393
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -38.0509    16.7820  -2.267  0.02337 *
## radius_mean                0.5227     0.4952   1.055  0.29120
## texture_mean               0.4617     0.1660   2.782  0.00540 **
## smoothness_mean            0.1289    99.8365   0.001  0.99897
## concavity_mean            58.3640    28.6695   2.036  0.04178 *
## symmetry_mean            -51.9111    37.6106  -1.380  0.16752
## fractal_dimension_mean   -87.0197   240.1909  -0.362  0.71713
## radius_se                 20.2642     7.6415   2.652  0.00801 **
## texture_se                 1.0909     1.2215   0.893  0.37184
## smoothness_se            289.9465   403.9299   0.718  0.47287
## compactness_se            56.8072   149.7642   0.379  0.70446
## concavity_se             -40.3851    44.4970  -0.908  0.36409
## concave_points_se       -100.3727   329.2203  -0.305  0.76046
## symmetry_se             -270.6388   157.0858  -1.723  0.08491 .
## fractal_dimension_se    -793.6856  1414.6402  -0.561  0.57476
## smoothness_worst          -5.8424    76.1926  -0.077  0.93888
## compactness_worst        -23.0579    20.7176  -1.113  0.26573
## concave_points_worst      74.0151    41.3274   1.791  0.07330 .
## symmetry_worst            70.7299    30.0863   2.351  0.01873 *
## fractal_dimension_worst   83.8086   196.5715   0.426  0.66985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 603.346  on 455  degrees of freedom
## Residual deviance:  47.131  on 436  degrees of freedom
## AIC: 87.131
##
## Number of Fisher Scoring iterations: 10
```

```r
pred <- predict(glm.fit, newdata = breast_test, type = "response")
y_test <- factor(breast_test$diagnosis)
auc_full <- auc(y_test, pred)
auc_full
```

```
## Area under the curve: 0.9946
```

### Newton-Raphson algorithm

### coordinate-wise optimization of a logistic-lasso model

```r
#soft threshold
sfxn <- function(beta, lambda) {
  if (abs(beta) > lambda) {
    return(sign(beta) * (abs(beta) - lambda))
```

```
  }
  else {
    return(0)
  }
}
```

```
#coordinate-wise optimization function
coordwise_lasso <- function(lambda, x, y, betastart, tol = exp(-10), maxiter = 5000) {
  i <- 0
  n <- length(y)
  pnum <- length(betastart)
  betavec <- betastart
  loglik <- 0
  res <- c(0, loglik, betavec)
  prevloglik <- -Inf
  while (i < maxiter & abs(loglik - prevloglik) > tol & loglik < Inf) {
    i <- i + 1
    prevloglik <- loglik
    for (j in 1:pnum) {
      theta <- x %*% betavec
      p <- exp(theta) / (1 + exp(theta)) #probability of malignant cases
      w <- p*(1-p) #working weights
      w <- ifelse(abs(w-0) < 1e-5, 1e-5, w)
      z <- theta + (y - p)/w #working response
      zwoj <- x[, -j] %*% betavec[-j]
      betavec[j] <- sfxn(sum(w*(x[,j])*(z - zwoj)), lambda) / (sum(w*x[,j]*x[,j]))
    }
    theta <- x %*% betavec
    p <- exp(theta) / (1 + exp(theta)) #probability of malignant cases
    w <- p*(1-p) #working weights
    w <- ifelse(abs(w-0) < 1e-10, 1e-10, w)
    z <- theta + (y - p)/w
    loglik <- sum(w*(z - theta)^2) / (2*n) + lambda * sum(abs(betavec))
    res <- rbind(res, c(i, loglik, betavec))
  }
  return(res)
}
#coordwise_res <- coordwise_lasso(lambda = 0.006, x_train_stan, y_train, betastart = rep(0, #20))
#coordwise_res[nrow(coordwise_res), ]
```

We need to calculate lambdamax first to define a sequence of lambda.

```
x.matrix <- scale(x_train) %>% as.matrix()
y.matrix <- as.matrix(y_train)
lambdamax <- max(abs(t(x.matrix) %*% y.matrix)) #/ nrow(y.matrix)
lambda_seq1 <- exp(seq(log(lambdamax), -5, length = 50))
lambda_seq2 <- exp(seq(log(lambdamax), -5, length = 50))
```

```
#a path of solutions
pathwise <- function(x, y, lambda) {
  n <- length(lambda)
  betastart <- rep(0, 20)
  betas <- NULL
```

```r
  for (i in 1:n) {
    coordwise_res <- coordwise_lasso(lambda = lambda[i],
                                     x = x,
                                     y = y,
                                     betastart = betastart)
    curbeta <- coordwise_res[nrow(coordwise_res), 3:22]
    betastart <- curbeta
    betas <- rbind(betas, c(curbeta))
  }
  return(data.frame(cbind(lambda, betas)))
}

pathwise_sol <- pathwise(x_train_stan, y_train, lambda_seq2)
round(pathwise_sol, 2) %>% knitr::kable()
```

| lambda | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 | V17 | V18 | V19 | V20 | V21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 176.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 143.31 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.30 | 0.00 | 0.00 |
| 116.45 | 0.00 | 0.05 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.52 | 0.00 | 0.00 |
| 94.62 | 0.00 | 0.19 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.67 | 0.00 | 0.00 |
| 76.88 | 0.00 | 0.32 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.81 | 0.00 | 0.00 |
| 62.47 | 0.00 | 0.46 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.95 | 0.00 | 0.00 |
| 50.75 | 0.00 | 0.61 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.09 | 0.00 | 0.00 |
| 41.24 | -0.05 | 0.74 | 0.04 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.21 | 0.00 | 0.00 |
| 33.51 | -0.13 | 0.82 | 0.12 | 0.00 | 0.00 | 0.00 | 0.00 | 0.05 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.33 | 0.00 | 0.00 |
| 27.23 | -0.20 | 0.90 | 0.19 | 0.00 | 0.00 | 0.00 | 0.00 | 0.15 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.42 | 0.02 | 0.00 |
| 22.12 | -0.26 | 0.99 | 0.27 | 0.00 | 0.00 | 0.00 | 0.00 | 0.26 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.47 | 0.10 | 0.00 |
| 17.97 | -0.31 | 1.09 | 0.34 | 0.00 | 0.00 | 0.00 | 0.00 | 0.37 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 1.52 | 0.17 | 0.00 |
| 14.61 | -0.35 | 1.23 | 0.42 | 0.00 | 0.00 | 0.00 | 0.00 | 0.48 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.08 | 0.00 | 1.52 | 0.23 | 0.00 |
| 11.87 | -0.38 | 1.38 | 0.50 | 0.00 | 0.00 | 0.00 | 0.00 | 0.58 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.15 | 0.00 | 1.54 | 0.29 | 0.00 |
| 9.64 | -0.42 | 1.53 | 0.58 | 0.00 | 0.00 | 0.00 | 0.00 | 0.68 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.22 | 0.00 | 1.56 | 0.35 | 0.00 |
| 7.83 | -0.45 | 1.65 | 0.66 | 0.00 | 0.00 | 0.00 | 0.00 | 0.82 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | -0.03 | 0.28 | 0.00 | 1.62 | 0.41 | 0.00 |
| 6.37 | -0.47 | 1.73 | 0.74 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | -0.09 | 0.32 | 0.00 | 1.71 | 0.46 | 0.00 |
| 5.17 | -0.49 | 1.82 | 0.82 | 0.00 | 0.03 | 0.00 | 0.00 | 1.15 | 0.00 | 0.00 | -0.02 | 0.00 | 0.00 | 0.00 | -0.15 | 0.37 | 0.00 | 1.80 | 0.51 | 0.00 |
| 4.20 | -0.49 | 1.90 | 0.91 | 0.00 | 0.20 | 0.00 | 0.00 | 1.29 | 0.00 | 0.00 | -0.15 | 0.00 | 0.00 | 0.00 | -0.17 | 0.39 | 0.00 | 1.85 | 0.56 | 0.00 |
| 3.41 | -0.49 | 2.00 | 1.00 | 0.00 | 0.36 | 0.00 | 0.00 | 1.43 | 0.00 | 0.00 | -0.28 | 0.00 | 0.00 | 0.00 | -0.19 | 0.41 | 0.00 | 1.90 | 0.61 | 0.00 |
| 2.77 | -0.49 | 2.11 | 1.10 | 0.00 | 0.53 | 0.00 | 0.00 | 1.57 | 0.00 | 0.00 | -0.40 | 0.00 | 0.00 | 0.00 | -0.21 | 0.45 | 0.00 | 1.95 | 0.66 | 0.00 |

| lambda | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 | V17 | V18 | V19 | V20 | V21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2.25 | -0.49 | 2.22 | 1.18 | 0.00 | 0.72 | 0.00 | 0.00 | 1.73 | 0.00 | 0.00 | -0.48 | 0.00 | 0.00 | -0.06 | -0.25 | 0.48 | 0.00 | 1.95 | 0.74 | 0.00 |
| 1.83 | -0.50 | 2.30 | 1.27 | 0.00 | 0.92 | 0.00 | -0.04 | 1.89 | 0.00 | 0.00 | -0.55 | 0.00 | 0.00 | -0.15 | -0.28 | 0.53 | 0.00 | 1.94 | 0.83 | 0.00 |
| 1.49 | -0.50 | 2.36 | 1.34 | 0.00 | 1.11 | 0.00 | -0.13 | 2.07 | 0.00 | 0.00 | -0.60 | 0.00 | 0.00 | -0.25 | -0.30 | 0.59 | 0.00 | 1.94 | 0.93 | 0.00 |
| 1.21 | -0.51 | 2.42 | 1.40 | 0.00 | 1.30 | 0.00 | -0.18 | 2.24 | 0.04 | 0.05 | -0.67 | 0.00 | 0.00 | -0.37 | -0.35 | 0.61 | 0.00 | 1.97 | 1.08 | 0.00 |
| 0.98 | -0.50 | 2.46 | 1.44 | 0.00 | 1.48 | -0.09 | -0.23 | 2.43 | 0.11 | 0.10 | -0.76 | -0.01 | 0.00 | -0.48 | -0.39 | 0.60 | 0.00 | 2.05 | 1.27 | 0.00 |
| 0.80 | -0.47 | 2.48 | 1.46 | 0.00 | 1.66 | -0.18 | -0.26 | 2.65 | 0.18 | 0.17 | -0.82 | -0.02 | -0.05 | -0.58 | -0.43 | 0.57 | 0.00 | 2.16 | 1.45 | 0.00 |
| 0.65 | -0.42 | 2.46 | 1.49 | 0.00 | 1.85 | -0.27 | -0.30 | 2.89 | 0.25 | 0.26 | -0.90 | -0.04 | -0.14 | -0.66 | -0.44 | 0.53 | 0.00 | 2.31 | 1.62 | 0.00 |
| 0.53 | -0.38 | 2.45 | 1.51 | 0.00 | 2.04 | -0.35 | -0.34 | 3.13 | 0.31 | 0.34 | -0.97 | -0.06 | -0.22 | -0.75 | -0.46 | 0.48 | 0.00 | 2.46 | 1.79 | 0.00 |
| 0.43 | -0.33 | 2.44 | 1.54 | 0.00 | 2.21 | -0.44 | -0.38 | 3.36 | 0.36 | 0.41 | -1.03 | -0.09 | -0.28 | -0.83 | -0.47 | 0.45 | 0.00 | 2.60 | 1.96 | 0.00 |
| 0.35 | -0.29 | 2.41 | 1.57 | 0.00 | 2.40 | -0.53 | -0.40 | 3.60 | 0.41 | 0.47 | -1.04 | -0.13 | -0.34 | -0.93 | -0.50 | 0.40 | -0.07 | 2.77 | 2.15 | 0.00 |
| 0.28 | -0.24 | 2.32 | 1.62 | 0.00 | 2.66 | -0.64 | -0.38 | 3.82 | 0.44 | 0.51 | -0.86 | -0.26 | -0.44 | -1.09 | -0.53 | 0.34 | -0.43 | 3.06 | 2.42 | 0.00 |
| 0.23 | -0.20 | 2.25 | 1.66 | 0.00 | 2.90 | -0.75 | -0.35 | 4.03 | 0.47 | 0.54 | -0.69 | -0.37 | -0.52 | -1.23 | -0.56 | 0.28 | -0.74 | 3.34 | 2.67 | 0.00 |
| 0.19 | -0.16 | 2.18 | 1.71 | 0.00 | 3.13 | -0.84 | -0.33 | 4.23 | 0.50 | 0.57 | -0.55 | -0.48 | -0.58 | -1.36 | -0.59 | 0.23 | -1.01 | 3.58 | 2.90 | 0.00 |
| 0.15 | -0.12 | 2.12 | 1.75 | -0.01 | 3.33 | -0.93 | -0.31 | 4.43 | 0.52 | 0.60 | -0.43 | -0.57 | -0.64 | -1.48 | -0.62 | 0.18 | -1.26 | 3.80 | 3.11 | 0.00 |
| 0.12 | -0.08 | 2.08 | 1.78 | -0.04 | 3.52 | -1.00 | -0.28 | 4.61 | 0.55 | 0.62 | -0.31 | -0.67 | -0.67 | -1.59 | -0.66 | 0.17 | -1.48 | 3.99 | 3.30 | 0.00 |
| 0.10 | -0.05 | 2.05 | 1.81 | -0.07 | 3.68 | -1.07 | -0.29 | 4.77 | 0.56 | 0.63 | -0.19 | -0.75 | -0.67 | -1.67 | -0.74 | 0.15 | -1.72 | 4.14 | 3.45 | 0.10 |
| 0.08 | -0.02 | 2.02 | 1.84 | -0.08 | 3.83 | -1.12 | -0.31 | 4.92 | 0.57 | 0.65 | -0.06 | -0.83 | -0.67 | -1.75 | -0.83 | 0.14 | -1.96 | 4.25 | 3.58 | 0.22 |
| 0.07 | 0.00 | 2.00 | 1.86 | -0.09 | 3.93 | -1.17 | -0.33 | 5.06 | 0.60 | 0.67 | -0.06 | -0.86 | -0.69 | -1.80 | -0.86 | 0.11 | -2.03 | 4.35 | 3.68 | 0.26 |
| 0.05 | 0.00 | 2.01 | 1.88 | -0.09 | 4.01 | -1.20 | -0.37 | 5.13 | 0.61 | 0.68 | 0.00 | -0.90 | -0.66 | -1.84 | -0.95 | 0.10 | -2.17 | 4.39 | 3.76 | 0.37 |
| 0.04 | 0.00 | 2.01 | 1.89 | -0.08 | 4.06 | -1.23 | -0.39 | 5.20 | 0.63 | 0.70 | 0.00 | -0.91 | -0.65 | -1.87 | -1.00 | 0.09 | -2.21 | 4.43 | 3.82 | 0.42 |
| 0.04 | 0.02 | 2.01 | 1.90 | -0.09 | 4.13 | -1.26 | -0.40 | 5.30 | 0.64 | 0.72 | 0.00 | -0.93 | -0.66 | -1.90 | -1.02 | 0.07 | -2.26 | 4.50 | 3.89 | 0.46 |
| 0.03 | 0.03 | 1.98 | 1.92 | -0.09 | 4.23 | -1.30 | -0.42 | 5.39 | 0.64 | 0.74 | 0.11 | -0.99 | -0.66 | -1.95 | -1.12 | 0.05 | -2.42 | 4.56 | 3.97 | 0.55 |
| 0.02 | 0.05 | 1.97 | 1.93 | -0.09 | 4.28 | -1.32 | -0.43 | 5.45 | 0.65 | 0.75 | 0.15 | -1.01 | -0.66 | -1.98 | -1.16 | 0.04 | -2.49 | 4.62 | 4.01 | 0.59 |
| 0.02 | 0.06 | 1.95 | 1.94 | -0.10 | 4.34 | -1.34 | -0.44 | 5.52 | 0.65 | 0.76 | 0.20 | -1.04 | -0.67 | -2.01 | -1.21 | 0.03 | -2.57 | 4.66 | 4.06 | 0.63 |
| 0.02 | 0.06 | 1.94 | 1.95 | -0.10 | 4.39 | -1.36 | -0.44 | 5.57 | 0.65 | 0.77 | 0.26 | -1.07 | -0.67 | -2.04 | -1.27 | 0.02 | -2.65 | 4.70 | 4.10 | 0.68 |
| 0.01 | 0.04 | 1.93 | 1.96 | -0.07 | 4.49 | -1.37 | -0.51 | 5.58 | 0.62 | 0.79 | 0.60 | -1.15 | -0.62 | -2.10 | -1.65 | 0.00 | -3.04 | 4.69 | 4.12 | 1.03 |

| lambda | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 | V17 | V18 | V19 | V20 | V21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.01 | 0.04 | 1.92 | 1.96 | -0.07 | 4.53 | -1.38 | -0.53 | 5.61 | 0.62 | 0.80 | 0.67 | -1.17 | -0.61 | -2.12 | -1.72 | 0.00 | -3.13 | 4.71 | 4.14 | 1.10 |
| 0.01 | 0.05 | 1.92 | 1.97 | -0.08 | 4.53 | -1.38 | -0.53 | 5.63 | 0.62 | 0.80 | 0.67 | -1.17 | -0.62 | -2.12 | -1.72 | 0.00 | -3.13 | 4.72 | 4.14 | 1.10 |
| 0.01 | 0.04 | 1.90 | 1.98 | -0.05 | 4.60 | -1.40 | -0.56 | 5.66 | 0.61 | 0.83 | 0.79 | -1.21 | -0.61 | -2.16 | -1.88 | -0.04 | -3.28 | 4.75 | 4.18 | 1.23 |

### cross-validation

```
set.seed(2022)

cv = function(data, lambda) {
  n <- nrow(data)
  data <- data[sample(n), ] #shuffle the data
  folds <- cut(seq(1, nrow(data)), breaks = 5, labels = FALSE) #Create 5 equal size folds
 # mse <- data.frame() #a data frame storing mse results
  #mse_lambda <- vector()
  #se <- vector() #a vector storing test errors
  res <- lambda
  #se <- vector() #a vectro storing test errors

    #Perform 5 fold cross validation
  for (i in 1:5) {
    #partition the data into train and test data
    testRows <- which(folds == i, arr.ind = TRUE)
    data_test <- data[testRows, ]
    data_train <- data[-testRows, ]
    x_train <- data_train[2:20]
    x_train_stan <- cbind(rep(1, nrow(x_train)), scale(x_train))
    y_train <- data_train[1]
    x_test <- data_test[2:20]
    #standardized test data
    x_test_stan <- cbind(rep(1, nrow(x_test)), scale(x_test))
    y_test <- data_test %>% mutate(diagnosis = factor(diagnosis))
    y_test <- y_test$diagnosis
    #Use the test and train data partitions to perform lasso
    path_sol <- pathwise(x = x_train_stan,
                         y = y_train,
                         lambda = lambda)
    auc <- vector()
    for (j in 1:length(lambda)) {
      curbeta <- as.numeric(path_sol[j, 2:21])
      theta <- x_test_stan %*% curbeta
      p <- exp(theta) / (1 + exp(theta))
      auc[j] <- auc(y_test, p)
      #y.pred <- ifelse(p > 0.5, 1, 0)
      #accuracy[j] <- mean(y.pred == y_test)
    }
    print(auc)
    res <- cbind(res, auc)
```

```r
    print(res)
  }
  return(res)
    #se[j] <- sqrt(var(error)/5)
  #cv.auc.lambda <- rowMeans(mse)
  #return(cv.auc.lambda)
}

cv_test = cv(data = breast_train, lambda_seq2)
```

```
##  [1] 0.5000000 0.5000000 0.9429688 0.9484375 0.9651042 0.9687500 0.9734375
##  [8] 0.9739583 0.9765625 0.9770833 0.9822917 0.9838542 0.9864583 0.9875000
## [15] 0.9880208 0.9911458 0.9911458 0.9911458 0.9911458 0.9916667 0.9921875
## [22] 0.9916667 0.9911458 0.9911458 0.9906250 0.9916667 0.9911458 0.9916667
## [29] 0.9927083 0.9927083 0.9932292 0.9927083 0.9916667 0.9916667 0.9916667
## [36] 0.9911458 0.9906250 0.9906250 0.9895833 0.9885417 0.9880208 0.9880208
## [43] 0.9875000 0.9875000 0.9864583 0.9864583 0.9864583 0.9864583 0.9864583
## [50] 0.9864583
##                res       auc
##  [1,] 1.763784e+02 0.5000000
##  [2,] 1.433124e+02 0.5000000
##  [3,] 1.164454e+02 0.9429688
##  [4,] 9.461516e+01 0.9484375
##  [5,] 7.687748e+01 0.9651042
##  [6,] 6.246511e+01 0.9687500
##  [7,] 5.075466e+01 0.9734375
##  [8,] 4.123959e+01 0.9739583
##  [9,] 3.350833e+01 0.9765625
## [10,] 2.722646e+01 0.9770833
## [11,] 2.212226e+01 0.9822917
## [12,] 1.797496e+01 0.9838542
## [13,] 1.460516e+01 0.9864583
## [14,] 1.186711e+01 0.9875000
## [15,] 9.642357e+00 0.9880208
## [16,] 7.834687e+00 0.9911458
## [17,] 6.365903e+00 0.9911458
## [18,] 5.172476e+00 0.9911458
## [19,] 4.202782e+00 0.9911458
## [20,] 3.414879e+00 0.9916667
## [21,] 2.774685e+00 0.9921875
## [22,] 2.254510e+00 0.9916667
## [23,] 1.831853e+00 0.9911458
## [24,] 1.488432e+00 0.9911458
## [25,] 1.209393e+00 0.9906250
## [26,] 9.826656e-01 0.9916667
## [27,] 7.984434e-01 0.9911458
## [28,] 6.487578e-01 0.9916667
## [29,] 5.271339e-01 0.9927083
## [30,] 4.283112e-01 0.9927083
## [31,] 3.480149e-01 0.9932292
## [32,] 2.827719e-01 0.9927083
## [33,] 2.297601e-01 0.9916667
## [34,] 1.866865e-01 0.9916667
```

```
## [35,] 1.516881e-01 0.9916667
## [36,] 1.232508e-01 0.9911458
## [37,] 1.001448e-01 0.9906250
## [38,] 8.137044e-02 0.9906250
## [39,] 6.611577e-02 0.9895833
## [40,] 5.372093e-02 0.9885417
## [41,] 4.364976e-02 0.9880208
## [42,] 3.546666e-02 0.9880208
## [43,] 2.881766e-02 0.9875000
## [44,] 2.341516e-02 0.9875000
## [45,] 1.902548e-02 0.9864583
## [46,] 1.545873e-02 0.9864583
## [47,] 1.256066e-02 0.9864583
## [48,] 1.020589e-02 0.9864583
## [49,] 8.292571e-03 0.9864583
## [50,] 6.737947e-03 0.9864583
##  [1] 0.5000000 0.5000000 0.9669625 0.9723866 0.9763314 0.9778107 0.9792899
##  [8] 0.9797830 0.9822485 0.9837278 0.9881657 0.9911243 0.9945759 0.9955621
## [15] 0.9965483 0.9965483 0.9980276 0.9990138 1.0000000 1.0000000 1.0000000
## [22] 1.0000000 1.0000000 0.9995069 0.9990138 0.9985207 0.9975345 0.9970414
## [29] 0.9960552 0.9955621 0.9945759 0.9945759 0.9945759 0.9916174 0.9911243
## [36] 0.9906312 0.9896450 0.9881657 0.9871795 0.9861933 0.9852071 0.9827416
## [43] 0.9827416 0.9827416 0.9817554 0.9817554 0.9817554 0.9817554 0.9817554
## [50] 0.9827416
##                res       auc       auc
##  [1,] 1.763784e+02 0.5000000 0.5000000
##  [2,] 1.433124e+02 0.5000000 0.5000000
##  [3,] 1.164454e+02 0.9429688 0.9669625
##  [4,] 9.461516e+01 0.9484375 0.9723866
##  [5,] 7.687748e+01 0.9651042 0.9763314
##  [6,] 6.246511e+01 0.9687500 0.9778107
##  [7,] 5.075466e+01 0.9734375 0.9792899
##  [8,] 4.123959e+01 0.9739583 0.9797830
##  [9,] 3.350833e+01 0.9765625 0.9822485
## [10,] 2.722646e+01 0.9770833 0.9837278
## [11,] 2.212226e+01 0.9822917 0.9881657
## [12,] 1.797496e+01 0.9838542 0.9911243
## [13,] 1.460516e+01 0.9864583 0.9945759
## [14,] 1.186711e+01 0.9875000 0.9955621
## [15,] 9.642357e+00 0.9880208 0.9965483
## [16,] 7.834687e+00 0.9911458 0.9965483
## [17,] 6.365903e+00 0.9911458 0.9980276
## [18,] 5.172476e+00 0.9911458 0.9990138
## [19,] 4.202782e+00 0.9911458 1.0000000
## [20,] 3.414879e+00 0.9916667 1.0000000
## [21,] 2.774685e+00 0.9921875 1.0000000
## [22,] 2.254510e+00 0.9916667 1.0000000
## [23,] 1.831853e+00 0.9911458 1.0000000
## [24,] 1.488432e+00 0.9911458 0.9995069
## [25,] 1.209393e+00 0.9906250 0.9990138
## [26,] 9.826656e-01 0.9916667 0.9985207
## [27,] 7.984434e-01 0.9911458 0.9975345
## [28,] 6.487578e-01 0.9916667 0.9970414
## [29,] 5.271339e-01 0.9927083 0.9960552
```

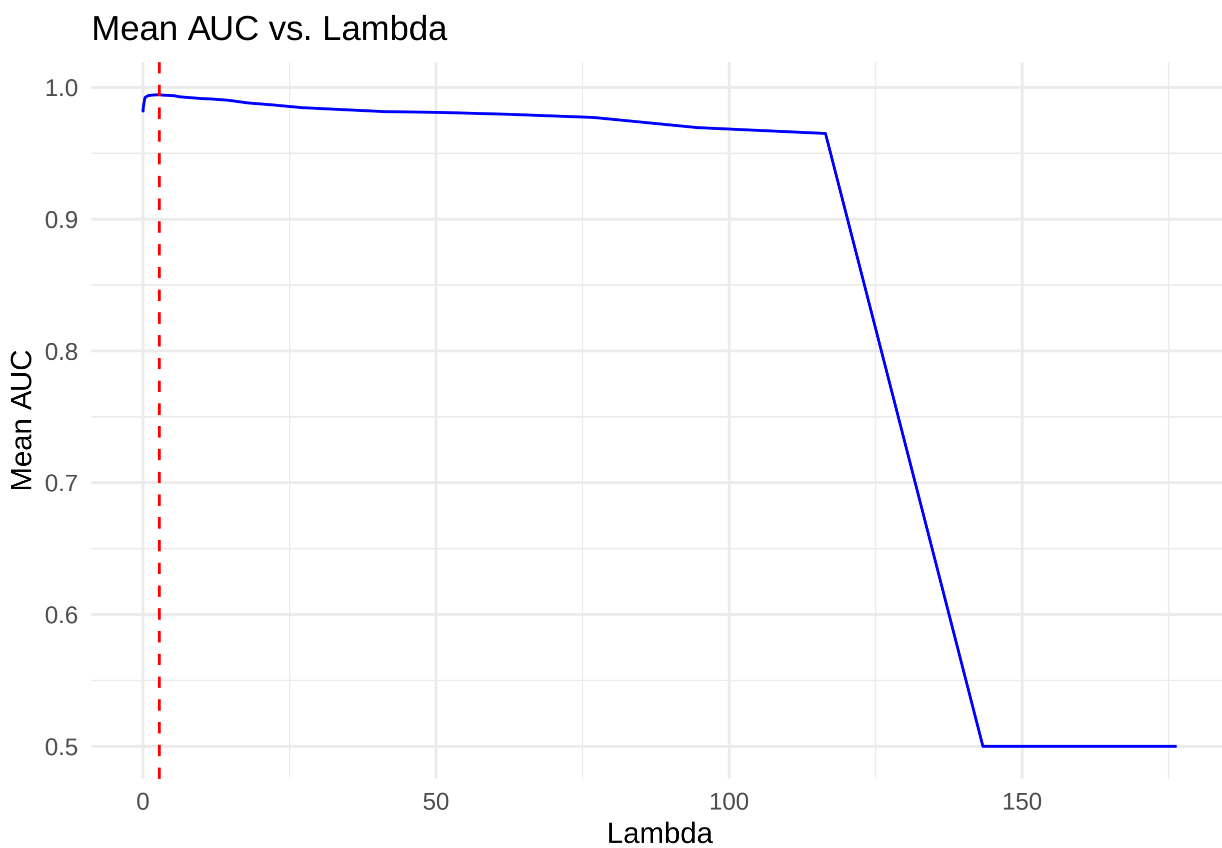
```
## [30,] 4.283112e-01 0.9927083 0.9955621
## [31,] 3.480149e-01 0.9932292 0.9945759
## [32,] 2.827719e-01 0.9927083 0.9945759
## [33,] 2.297601e-01 0.9916667 0.9945759
## [34,] 1.866865e-01 0.9916667 0.9916174
## [35,] 1.516881e-01 0.9916667 0.9911243
## [36,] 1.232508e-01 0.9911458 0.9906312
## [37,] 1.001448e-01 0.9906250 0.9896450
## [38,] 8.137044e-02 0.9906250 0.9881657
## [39,] 6.611577e-02 0.9895833 0.9871795
## [40,] 5.372093e-02 0.9885417 0.9861933
## [41,] 4.364976e-02 0.9880208 0.9852071
## [42,] 3.546666e-02 0.9880208 0.9827416
## [43,] 2.881766e-02 0.9875000 0.9827416
## [44,] 2.341516e-02 0.9875000 0.9827416
## [45,] 1.902548e-02 0.9864583 0.9817554
## [46,] 1.545873e-02 0.9864583 0.9817554
## [47,] 1.256066e-02 0.9864583 0.9817554
## [48,] 1.020589e-02 0.9864583 0.9817554
## [49,] 8.292571e-03 0.9864583 0.9817554
## [50,] 6.737947e-03 0.9864583 0.9827416
##  [1] 0.5000000 0.5000000 0.9644049 0.9655172 0.9733037 0.9755284 0.9760845
##  [8] 0.9760845 0.9766407 0.9783092 0.9777531 0.9799778 0.9810901 0.9822024
## [15] 0.9827586 0.9833148 0.9844271 0.9877642 0.9877642 0.9883204 0.9888765
## [22] 0.9894327 0.9894327 0.9894327 0.9894327 0.9894327 0.9883204 0.9877642
## [29] 0.9877642 0.9872080 0.9877642 0.9888765 0.9883204 0.9877642 0.9877642
## [36] 0.9860957 0.9855395 0.9855395 0.9866518 0.9866518 0.9855395 0.9860957
## [43] 0.9860957 0.9849833 0.9855395 0.9827586 0.9827586 0.9827586 0.9827586
## [50] 0.9827586
##                res       auc       auc       auc
##  [1,] 1.763784e+02 0.5000000 0.5000000 0.5000000
##  [2,] 1.433124e+02 0.5000000 0.5000000 0.5000000
##  [3,] 1.164454e+02 0.9429688 0.9669625 0.9644049
##  [4,] 9.461516e+01 0.9484375 0.9723866 0.9655172
##  [5,] 7.687748e+01 0.9651042 0.9763314 0.9733037
##  [6,] 6.246511e+01 0.9687500 0.9778107 0.9755284
##  [7,] 5.075466e+01 0.9734375 0.9792899 0.9760845
##  [8,] 4.123959e+01 0.9739583 0.9797830 0.9760845
##  [9,] 3.350833e+01 0.9765625 0.9822485 0.9766407
## [10,] 2.722646e+01 0.9770833 0.9837278 0.9783092
## [11,] 2.212226e+01 0.9822917 0.9881657 0.9777531
## [12,] 1.797496e+01 0.9838542 0.9911243 0.9799778
## [13,] 1.460516e+01 0.9864583 0.9945759 0.9810901
## [14,] 1.186711e+01 0.9875000 0.9955621 0.9822024
## [15,] 9.642357e+00 0.9880208 0.9965483 0.9827586
## [16,] 7.834687e+00 0.9911458 0.9965483 0.9833148
## [17,] 6.365903e+00 0.9911458 0.9980276 0.9844271
## [18,] 5.172476e+00 0.9911458 0.9990138 0.9877642
## [19,] 4.202782e+00 0.9911458 1.0000000 0.9877642
## [20,] 3.414879e+00 0.9916667 1.0000000 0.9883204
## [21,] 2.774685e+00 0.9921875 1.0000000 0.9888765
## [22,] 2.254510e+00 0.9916667 1.0000000 0.9894327
## [23,] 1.831853e+00 0.9911458 1.0000000 0.9894327
## [24,] 1.488432e+00 0.9911458 0.9995069 0.9894327
```

```
## [25,] 1.209393e+00 0.9906250 0.9990138 0.9894327
## [26,] 9.826656e-01 0.9916667 0.9985207 0.9894327
## [27,] 7.984434e-01 0.9911458 0.9975345 0.9883204
## [28,] 6.487578e-01 0.9916667 0.9970414 0.9877642
## [29,] 5.271339e-01 0.9927083 0.9960552 0.9877642
## [30,] 4.283112e-01 0.9927083 0.9955621 0.9872080
## [31,] 3.480149e-01 0.9932292 0.9945759 0.9877642
## [32,] 2.827719e-01 0.9927083 0.9945759 0.9888765
## [33,] 2.297601e-01 0.9916667 0.9945759 0.9883204
## [34,] 1.866865e-01 0.9916667 0.9916174 0.9877642
## [35,] 1.516881e-01 0.9916667 0.9911243 0.9877642
## [36,] 1.232508e-01 0.9911458 0.9906312 0.9860957
## [37,] 1.001448e-01 0.9906250 0.9896450 0.9855395
## [38,] 8.137044e-02 0.9906250 0.9881657 0.9855395
## [39,] 6.611577e-02 0.9895833 0.9871795 0.9866518
## [40,] 5.372093e-02 0.9885417 0.9861933 0.9866518
## [41,] 4.364976e-02 0.9880208 0.9852071 0.9855395
## [42,] 3.546666e-02 0.9880208 0.9827416 0.9860957
## [43,] 2.881766e-02 0.9875000 0.9827416 0.9860957
## [44,] 2.341516e-02 0.9875000 0.9827416 0.9849833
## [45,] 1.902548e-02 0.9864583 0.9817554 0.9855395
## [46,] 1.545873e-02 0.9864583 0.9817554 0.9827586
## [47,] 1.256066e-02 0.9864583 0.9817554 0.9827586
## [48,] 1.020589e-02 0.9864583 0.9817554 0.9827586
## [49,] 8.292571e-03 0.9864583 0.9817554 0.9827586
## [50,] 6.737947e-03 0.9864583 0.9827416 0.9827586
##  [1] 0.5000000 0.5000000 0.9892857 0.9933673 0.9954082 0.9964286 0.9964286
##  [8] 0.9964286 0.9964286 0.9974490 0.9979592 0.9979592 0.9989796 0.9994898
## [15] 0.9994898 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
## [22] 1.0000000 1.0000000 1.0000000 1.0000000 0.9994898 0.9994898 0.9994898
## [29] 0.9994898 0.9994898 0.9994898 0.9994898 0.9994898 0.9994898 0.9989796
## [36] 0.9989796 0.9989796 0.9989796 0.9989796 0.9984694 0.9984694 0.9984694
## [43] 0.9984694 0.9984694 0.9984694 0.9984694 0.9984694 0.9984694 0.9984694
## [50] 0.9984694
##                res       auc       auc       auc       auc
##  [1,] 1.763784e+02 0.5000000 0.5000000 0.5000000 0.5000000
##  [2,] 1.433124e+02 0.5000000 0.5000000 0.5000000 0.5000000
##  [3,] 1.164454e+02 0.9429688 0.9669625 0.9644049 0.9892857
##  [4,] 9.461516e+01 0.9484375 0.9723866 0.9655172 0.9933673
##  [5,] 7.687748e+01 0.9651042 0.9763314 0.9733037 0.9954082
##  [6,] 6.246511e+01 0.9687500 0.9778107 0.9755284 0.9964286
##  [7,] 5.075466e+01 0.9734375 0.9792899 0.9760845 0.9964286
##  [8,] 4.123959e+01 0.9739583 0.9797830 0.9760845 0.9964286
##  [9,] 3.350833e+01 0.9765625 0.9822485 0.9766407 0.9964286
## [10,] 2.722646e+01 0.9770833 0.9837278 0.9783092 0.9974490
## [11,] 2.212226e+01 0.9822917 0.9881657 0.9777531 0.9979592
## [12,] 1.797496e+01 0.9838542 0.9911243 0.9799778 0.9979592
## [13,] 1.460516e+01 0.9864583 0.9945759 0.9810901 0.9989796
## [14,] 1.186711e+01 0.9875000 0.9955621 0.9822024 0.9994898
## [15,] 9.642357e+00 0.9880208 0.9965483 0.9827586 0.9994898
## [16,] 7.834687e+00 0.9911458 0.9965483 0.9833148 1.0000000
## [17,] 6.365903e+00 0.9911458 0.9980276 0.9844271 1.0000000
## [18,] 5.172476e+00 0.9911458 0.9990138 0.9877642 1.0000000
## [19,] 4.202782e+00 0.9911458 1.0000000 0.9877642 1.0000000
```

```
## [20,] 3.414879e+00 0.9916667 1.0000000 0.9883204 1.0000000
## [21,] 2.774685e+00 0.9921875 1.0000000 0.9888765 1.0000000
## [22,] 2.254510e+00 0.9916667 1.0000000 0.9894327 1.0000000
## [23,] 1.831853e+00 0.9911458 1.0000000 0.9894327 1.0000000
## [24,] 1.488432e+00 0.9911458 0.9995069 0.9894327 1.0000000
## [25,] 1.209393e+00 0.9906250 0.9990138 0.9894327 1.0000000
## [26,] 9.826656e-01 0.9916667 0.9985207 0.9894327 0.9994898
## [27,] 7.984434e-01 0.9911458 0.9975345 0.9883204 0.9994898
## [28,] 6.487578e-01 0.9916667 0.9970414 0.9877642 0.9994898
## [29,] 5.271339e-01 0.9927083 0.9960552 0.9877642 0.9994898
## [30,] 4.283112e-01 0.9927083 0.9955621 0.9872080 0.9994898
## [31,] 3.480149e-01 0.9932292 0.9945759 0.9877642 0.9994898
## [32,] 2.827719e-01 0.9927083 0.9945759 0.9888765 0.9994898
## [33,] 2.297601e-01 0.9916667 0.9945759 0.9883204 0.9994898
## [34,] 1.866865e-01 0.9916667 0.9916174 0.9877642 0.9994898
## [35,] 1.516881e-01 0.9916667 0.9911243 0.9877642 0.9989796
## [36,] 1.232508e-01 0.9911458 0.9906312 0.9860957 0.9989796
## [37,] 1.001448e-01 0.9906250 0.9896450 0.9855395 0.9989796
## [38,] 8.137044e-02 0.9906250 0.9881657 0.9855395 0.9989796
## [39,] 6.611577e-02 0.9895833 0.9871795 0.9866518 0.9989796
## [40,] 5.372093e-02 0.9885417 0.9861933 0.9866518 0.9984694
## [41,] 4.364976e-02 0.9880208 0.9852071 0.9855395 0.9984694
## [42,] 3.546666e-02 0.9880208 0.9827416 0.9860957 0.9984694
## [43,] 2.881766e-02 0.9875000 0.9827416 0.9860957 0.9984694
## [44,] 2.341516e-02 0.9875000 0.9827416 0.9849833 0.9984694
## [45,] 1.902548e-02 0.9864583 0.9817554 0.9855395 0.9984694
## [46,] 1.545873e-02 0.9864583 0.9817554 0.9827586 0.9984694
## [47,] 1.256066e-02 0.9864583 0.9817554 0.9827586 0.9984694
## [48,] 1.020589e-02 0.9864583 0.9817554 0.9827586 0.9984694
## [49,] 8.292571e-03 0.9864583 0.9817554 0.9827586 0.9984694
## [50,] 6.737947e-03 0.9864583 0.9827416 0.9827586 0.9984694
##  [1] 0.5000000 0.5000000 0.9616162 0.9676768 0.9757576 0.9792929 0.9797980
##  [8] 0.9818182 0.9843434 0.9863636 0.9873737 0.9878788 0.9898990 0.9909091
## [15] 0.9914141 0.9904040 0.9904040 0.9909091 0.9909091 0.9904040 0.9909091
## [22] 0.9904040 0.9904040 0.9904040 0.9904040 0.9909091 0.9909091 0.9898990
## [29] 0.9883838 0.9873737 0.9868687 0.9803030 0.9772727 0.9732323 0.9686869
## [36] 0.9676768 0.9676768 0.9676768 0.9656566 0.9656566 0.9656566 0.9656566
## [43] 0.9656566 0.9651515 0.9646465 0.9646465 0.9646465 0.9646465 0.9611111
## [50] 0.9611111
##                res       auc       auc       auc       auc       auc
##  [1,] 1.763784e+02 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000
##  [2,] 1.433124e+02 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000
##  [3,] 1.164454e+02 0.9429688 0.9669625 0.9644049 0.9892857 0.9616162
##  [4,] 9.461516e+01 0.9484375 0.9723866 0.9655172 0.9933673 0.9676768
##  [5,] 7.687748e+01 0.9651042 0.9763314 0.9733037 0.9954082 0.9757576
##  [6,] 6.246511e+01 0.9687500 0.9778107 0.9755284 0.9964286 0.9792929
##  [7,] 5.075466e+01 0.9734375 0.9792899 0.9760845 0.9964286 0.9797980
##  [8,] 4.123959e+01 0.9739583 0.9797830 0.9760845 0.9964286 0.9818182
##  [9,] 3.350833e+01 0.9765625 0.9822485 0.9766407 0.9964286 0.9843434
## [10,] 2.722646e+01 0.9770833 0.9837278 0.9783092 0.9974490 0.9863636
## [11,] 2.212226e+01 0.9822917 0.9881657 0.9777531 0.9979592 0.9873737
## [12,] 1.797496e+01 0.9838542 0.9911243 0.9799778 0.9979592 0.9878788
## [13,] 1.460516e+01 0.9864583 0.9945759 0.9810901 0.9989796 0.9898990
## [14,] 1.186711e+01 0.9875000 0.9955621 0.9822024 0.9994898 0.9909091
```

```
## [15,] 9.642357e+00 0.9880208 0.9965483 0.9827586 0.9994898 0.9914141
## [16,] 7.834687e+00 0.9911458 0.9965483 0.9833148 1.0000000 0.9904040
## [17,] 6.365903e+00 0.9911458 0.9980276 0.9844271 1.0000000 0.9904040
## [18,] 5.172476e+00 0.9911458 0.9990138 0.9877642 1.0000000 0.9909091
## [19,] 4.202782e+00 0.9911458 1.0000000 0.9877642 1.0000000 0.9909091
## [20,] 3.414879e+00 0.9916667 1.0000000 0.9883204 1.0000000 0.9904040
## [21,] 2.774685e+00 0.9921875 1.0000000 0.9888765 1.0000000 0.9909091
## [22,] 2.254510e+00 0.9916667 1.0000000 0.9894327 1.0000000 0.9904040
## [23,] 1.831853e+00 0.9911458 1.0000000 0.9894327 1.0000000 0.9904040
## [24,] 1.488432e+00 0.9911458 0.9995069 0.9894327 1.0000000 0.9904040
## [25,] 1.209393e+00 0.9906250 0.9990138 0.9894327 1.0000000 0.9904040
## [26,] 9.826656e-01 0.9916667 0.9985207 0.9894327 0.9994898 0.9909091
## [27,] 7.984434e-01 0.9911458 0.9975345 0.9883204 0.9994898 0.9909091
## [28,] 6.487578e-01 0.9916667 0.9970414 0.9877642 0.9994898 0.9898990
## [29,] 5.271339e-01 0.9927083 0.9960552 0.9877642 0.9994898 0.9883838
## [30,] 4.283112e-01 0.9927083 0.9955621 0.9872080 0.9994898 0.9873737
## [31,] 3.480149e-01 0.9932292 0.9945759 0.9877642 0.9994898 0.9868687
## [32,] 2.827719e-01 0.9927083 0.9945759 0.9888765 0.9994898 0.9803030
## [33,] 2.297601e-01 0.9916667 0.9945759 0.9883204 0.9994898 0.9772727
## [34,] 1.866865e-01 0.9916667 0.9916174 0.9877642 0.9994898 0.9732323
## [35,] 1.516881e-01 0.9916667 0.9911243 0.9877642 0.9989796 0.9686869
## [36,] 1.232508e-01 0.9911458 0.9906312 0.9860957 0.9989796 0.9676768
## [37,] 1.001448e-01 0.9906250 0.9896450 0.9855395 0.9989796 0.9676768
## [38,] 8.137044e-02 0.9906250 0.9881657 0.9855395 0.9989796 0.9676768
## [39,] 6.611577e-02 0.9895833 0.9871795 0.9866518 0.9989796 0.9656566
## [40,] 5.372093e-02 0.9885417 0.9861933 0.9866518 0.9984694 0.9656566
## [41,] 4.364976e-02 0.9880208 0.9852071 0.9855395 0.9984694 0.9656566
## [42,] 3.546666e-02 0.9880208 0.9827416 0.9860957 0.9984694 0.9656566
## [43,] 2.881766e-02 0.9875000 0.9827416 0.9860957 0.9984694 0.9656566
## [44,] 2.341516e-02 0.9875000 0.9827416 0.9849833 0.9984694 0.9651515
## [45,] 1.902548e-02 0.9864583 0.9817554 0.9855395 0.9984694 0.9646465
## [46,] 1.545873e-02 0.9864583 0.9817554 0.9827586 0.9984694 0.9646465
## [47,] 1.256066e-02 0.9864583 0.9817554 0.9827586 0.9984694 0.9646465
## [48,] 1.020589e-02 0.9864583 0.9817554 0.9827586 0.9984694 0.9646465
## [49,] 8.292571e-03 0.9864583 0.9817554 0.9827586 0.9984694 0.9611111
## [50,] 6.737947e-03 0.9864583 0.9827416 0.9827586 0.9984694 0.9611111
```

```r
cv_res <- as.data.frame(cv_test) #colnames(c("auc1", "auc2", "auc3", "auc4", "auc5"))
colnames(cv_res) <- c("res", "auc1", "auc2", "auc3", "auc4", "auc5")
cv_lambda <- cv_res[1]
mean_auc <- cv_res %>% dplyr::select(-1) %>% rowMeans()
cv_auc <- cbind(cv_lambda, mean_auc)
maxauc <- max(cv_auc$mean_auc)
bestlambda <- cv_auc[which(cv_auc$mean_auc == maxauc ),]$res
cv_auc %>%
  ggplot(x = res, y = mean_auc ) +
  geom_line(aes(x = res, y = mean_auc), col = "blue") +
  geom_vline(xintercept = bestlambda, linetype = "dashed", col = "red") +
  labs(title = "Mean AUC vs. Lambda",
       x = "Lambda",
       y = "Mean AUC")
```




### Compare full model and lasso model

```r
#corresponding betas of best lambda
lasso_beta <- pathwise_sol[which(pathwise_sol$lambda == bestlambda ),][2:21] %>% as.numeric()

#prediction performance function
predict <- function(x, y, betavec) {
  theta <- x %*% betavec
  p <- exp(theta) / (1 + exp(theta))
  auc <- auc(y, p)
}

auc_lasso <- predict(x_test_stan, y_test, lasso_beta)
auc_lasso
```

```
## Area under the curve: 0.9932
```

```r
cbind(auc_full, auc_lasso) %>% knitr::kable()
```

| auc_full | auc_lasso |
|---|---|
| 0.9945799 | 0.9932249 |

```r
#coefficients of full and lasso models
glm_beta <- glm.fit$coefficients %>% as.vector()
coefnames <- rownames(coef(summary(glm.fit)))
cbind(coefnames, glm_beta, lasso_beta) %>% knitr::kable()
```

| coefnames | glm_beta | lasso_beta |
|---|---|---|
| (Intercept) | -38.0508879315254 | -0.491575282868433 |
| radius_mean | 0.522652718986337 | 2.1061341199202 |
| texture_mean | 0.461723878543842 | 1.09553982152909 |
| smoothness_mean | 0.1289206136156 | 0 |
| concavity_mean | 58.3639721687468 | 0.531755692852326 |
| symmetry_mean | -51.9110605975334 | 0 |
| fractal_dimension_mean | -87.0196726852845 | 0 |
| radius_se | 20.2642224845343 | 1.57060392035703 |
| texture_se | 1.09086865732585 | 0 |
| smoothness_se | 289.946465559794 | 0 |
| compactness_se | 56.8072085913126 | -0.39710032709783 |
| concavity_se | -40.3851421619712 | 0 |
| concave_points_se | -100.372695041936 | 0 |
| symmetry_se | -270.638750016476 | 0 |
| fractal_dimension_se | -793.685579711044 | -0.214263582015782 |
| smoothness_worst | -5.84243418974478 | 0.446429427867803 |
| compactness_worst | -23.0578666420758 | 0 |
| concave_points_worst | 74.0151310096505 | 1.95085661565962 |
| symmetry_worst | 70.7298970265517 | 0.662947487333 |
| fractal_dimension_worst | 83.8086140810761 | 0 |