logistic

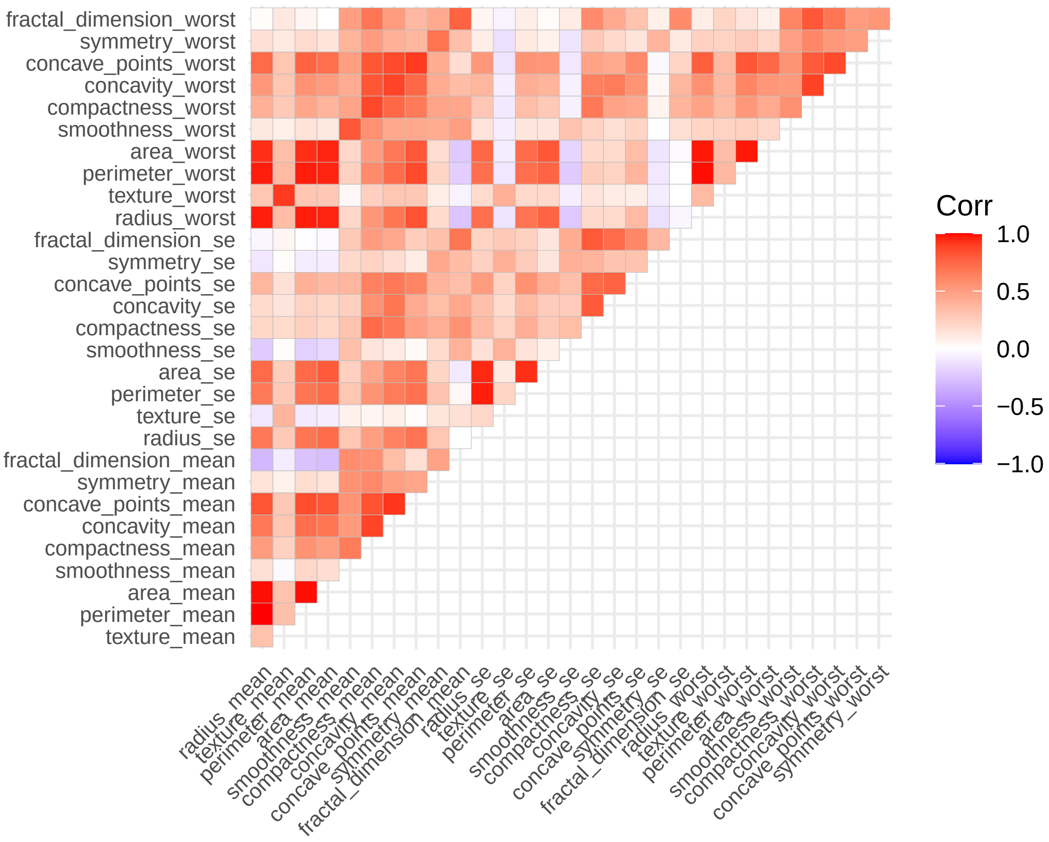
Xinran Sun, Haotian Wu, Lin Yang, Shengzhi Luo

3/17/2022

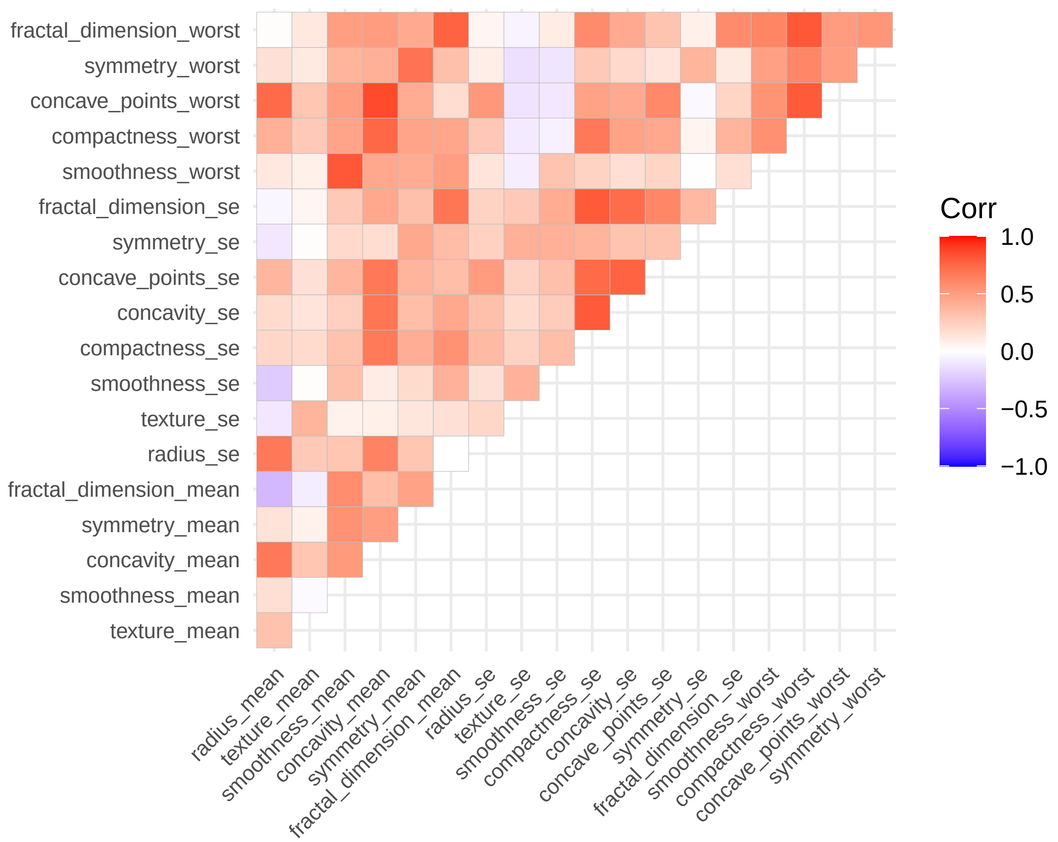
```
ggplot2::theme_set(theme_minimal() + theme(legend.position = "bottom"))
```

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
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
```

```r
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

```r
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
## -1.56866  -0.00003   0.00000   0.00000   2.82897
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -85.8416    36.2655  -2.367   0.0179 *
## radius_mean                0.4828     0.5136   0.940   0.3472
## texture_mean               1.2278     0.4834   2.540   0.0111 *
## smoothness_mean          -61.2848   138.3795  -0.443   0.6579
## concavity_mean           277.1480   130.0326   2.131   0.0331 *
## symmetry_mean           -174.8730    92.2200  -1.896   0.0579 .
## fractal_dimension_mean  -268.5035   380.8394  -0.705   0.4808
## radius_se                 68.8555    28.0557   2.454   0.0141 *
## texture_se                -0.4983     2.0041  -0.249   0.8036
## smoothness_se           1374.2231   645.1713   2.130   0.0332 *
## compactness_se            42.7437   215.7816   0.198   0.8430
## concavity_se            -150.6774    83.3050  -1.809   0.0705 .
## concave_points_se        731.2978   563.0794   1.299   0.1940
## symmetry_se            -1308.6896   653.0464  -2.004   0.0451 *
## fractal_dimension_se   -6434.2765  2965.7057  -2.170   0.0300 *
## smoothness_worst        -133.9879    98.5470  -1.360   0.1739
## compactness_worst       -104.6221    49.1751  -2.128   0.0334 *
## concave_points_worst     161.9938    89.1095   1.818   0.0691 .
## symmetry_worst           180.8713    87.6875   2.063   0.0391 *
## fractal_dimension_worst  807.8439   356.3095   2.267   0.0234 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 599.11  on 455  degrees of freedom
## Residual deviance:  27.13  on 436  degrees of freedom
## AIC: 67.13
##
## Number of Fisher Scoring iterations: 13
```

```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: 0.9771
```

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
```

```
  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)
pathwise_sol %>% knitr::kable()
```

lambda	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21
[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000
[illegible]	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000
[illegible]	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000
[illegible]	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000
[illegible]	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000
50.4808396	-0.0278073	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000
41.0222632	-0.1251272	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000
33.3359367	-0.2096374	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000
27.0897945	-0.2869031	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000
22.0139896	-0.3559700	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	[illegible]
17.8892364	-0.4187605	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	[illegible]	[illegible]
14.5373368	-0.4711694	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
11.8134813	-0.5191534	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
9.5999936	-0.5631970	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
7.8012463	-0.5940173	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	-0.0649553	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
6.3395297	-0.6195185	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	-0.1338648	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
5.1516943	-0.6407076	[illegible]	[illegible]	[illegible]	0.0000000	0.0000000	0.0000000	[illegible]	[illegible]	0.0000000	-0.0047169	0.0000000	0.0000000	0.0000000	-0.2025796	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4.1864232	-0.6542982	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.0000000	[illegible]	[illegible]	0.0000000	-0.1015655	0.0000000	0.0000000	0.0000000	-0.2154693	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
3.4020147	-0.6500881	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.0000000	[illegible]	[illegible]	0.0000000	-0.2380164	0.0000000	0.0000000	0.0000000	-0.2478217	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2.7645804	-0.6457690	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.0000000	[illegible]	[illegible]	0.0000000	-0.3330720	0.0000000	0.0000000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

lambda V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21

2.2465820 2.0159.4028.1.4620.0675.2140000000.0969.0440000000-00 0.0000000000 - 0.4390700020010894686.8290000
0.6396141 0.4290305 0.2614630256765

1.8256408 2.0795.0254.0.5900.0915.8943000000.2026.6137000000-00 0.0000000000 - 0.4642324002010407485.4030000
0.6318917 0.5274154 0.3851086214642

1.4835711 2.1362.6376.1950.1016.5464200000 2.5045989000000931-95 0.0000000000 - 0.4840300020168.5628.9900000
0.6247137 0.0094589 0.6343701 0.5100948188043

1.2055949 2.1101.3932.1001.0408141-25 0.0000207014864000126234-1 0.0000000000 - 0.3934078500202453.9585.3010000
0.6204545 0.0334200 0.8354201 0.6937095223518

0.9797031 2.0711.6968.4.1002.0.1081017-00 0.0000301092458000233535-1 0.0000000000 - 0.3179598500203717.1679.0250000
0.6105199 0.1273622 1.0432926 0.8585087819011

0.7961365 2.0583.5243.7.3004.2016786-61 - 3.452274.1000378360-9 0.0000000000 - 0.1873850800207.2.9206.1738494
0.5713337 0.192209472910 1.3045021 0.9976481287000

0.6469647 2.0331.6779.1.2080.2.0472363 - 3.8094806000050213532 0.0000000000 - 0.0628924500205714.2950.4056093
0.5263515 0.2512391759713 1.5780634 1.1253296437735

0.5257431 2.0501.7694.0.5650.2080.11989 - 4.16798300000622695-77 0.0000000000 - 0.000000000020657.5498.5855882
0.4818089 0.3210050951391 1.8072508 1.2366125911216

0.4272348 2.1002.8737.4.0003.0220.5446 - 4.524483100006080912 0.0000000000 - 0.0000000000207202.5423.3751544
0.4330860 0.3977025082752 2.0154409 1.3189192729992

0.3471840 2.1261.6889.7.2570.3.0501.3429 - 4.882830490000704061906 - 0.0000000 - 0.0000000 2.8631.6981.3.0609793
0.3821628 0.4975054135814 2.0554074508868 1.4446173526781 0.2833424

0.2821323 2.1125.3280.7.0870.4005.0954 - 5.255709550007744717 - 0.0000000 - 0.0000000 3.0856.8089.6586585
0.3243007 0.6292058215336 1.911004.3792406 1.6259127815449 0.8669767

0.2292692 2.0228.0.1478.2.0040.6125630 - 5.672635000080972601 - 0.0000000 - - - 3.4012.4821.2.0400293
0.2789284 0.7560460349233 1.7930273241607 1.8891094883552744811362

0.1863111 1.8912.5102.6.6760.5.0273611-5 - 6.0999290730010834611 - 0.0000000 - - - 3.7782.1589.5580328
0.2516542 0.9015081833770 1.5573055388667 2.1981958098369.0527202460

0.1514021 1.7582.2564.1.5400.6.000921-05 - 6.5582941000800388451 - 0.0000000 - - - 4.1892.9230.2377305
0.2350067 1.0618876107694 1.2790237525245 2.5345926010325737989577

0.1230339 1.6521.3698469 6.8421318 - 7.0694318 1.4917187 - 0.2013255 - - - 4.4513.0356.9183653
0.2436290 0.0326285 1.2354098279707 0.0282237 1.0660.197882783 2.9336450109873378214850

0.0999811 1.5722.0186.5312 7.7114249 - 7.6280353 1.6718872 - 0.4802175 - - - 4.6813.6774.5827317
0.2632320 0.1196103 1.4142680062344 0.0624598 0.9356724278185 3.3564985947994.0418534321

0.0812477 1.5841.1532.0271 7.8581896 - 7.8442642 1.6917611 - 0.5535705 - - - 4.8213.8114.6867455
0.2704924 0.1502201 1.4850108576607 0.0633106 1.0301527855367 3.4311419438476449591522

0.0660243 1.4847.7842797 9.2211121 - 8.7177878 2.0153037 - 0.8918518 - - - 5.2280.8255.6282987
0.3012377 0.2562806 1.7473539593630 0.1016379 0.9247.8613.2698 4.104826527331216.5837296

0.0536534 1.4663.5455898 10.5188493 - 9.5640994 2.2571779 - 1.5149891 - - - 5.5183.3226.8146122
0.4077028 0.3606266 2.0536130270349 0.1589325 0.2962.1005.8805 4.80067820.9154922728172269

0.0436004 1.4647.3692.5232 11.4471155 - 10.27662662.4556677 - 1.8221497 - - - 5.8945.6347.6962438
0.4585712 0.4194876 2.2845.1900.213792 0.1819305 0.1258236496368 5.2928518412837.2856074

0.0354310 1.4730.3423786 12.3706586 - 10.99251322.6457.4020000 2.1116853 - - - 6.2886.1338.4032452
0.5058222 0.4729787 2.5153371586186 0.2007129 2.5883931 5.7809186005725789509484

0.0287923 1.4976.5323913 13.2834056 - 11.68775192.8279.2030000 2.3759244 - - - 6.6716.8729.7310521
0.5449859 0.5239582 2.7356622841103 0.2108514 2.7911489 6.2529185186489089754568

0.0233975 1.5224.4672.7267 14.1759302 - 12.36539773.0005.1000000 2.6241298 - - - 7.0567.0970.6319670
0.5820266 0.5673982 2.9508182037531 0.2189496 2.9865965 6.71631873302728012241219615

0.0190135 1.5102.4398477 14.2492477 - 12.47672433.0182.6100000 2.6287574 - - - 7.1587.4943.2730981
0.5876231 0.5733327 2.9856531.063809 0.2238884 3.0019211 6.76513.3287.2048.2539892752

0.0154510 1.5604.2083.6737 15.8530908 - 13.63863093.3117.3201.7067 3.0818013 - - - 7.8107.9975.9597713
0.6547548 0.6348092 3.3603402276422 0.2349384 3.3707509 7.5992129712798510535923

0.0125559 1.5714.1887.2212 16.6117316 - 14.21979263.4474.0228.3471 3.2923269 - - - 8.1689.4710.1836096171
0.6915566 0.6617006 3.5507750760908 0.2451917 3.5616824 8.006103480437488299084858

lambda	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21
0.0102033	-0.7248177	[illegible]	[illegible]	-0.6859605	17.3113466	[illegible]	[illegible]	[illegible]	-0.2534951	[illegible]	[illegible]	-3.7342582	3.4837598	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
0.0082915	-0.7545704	[illegible]	[illegible]	-0.7076959	17.9474420	[illegible]	[illegible]	[illegible]	-0.2601638	[illegible]	[illegible]	-3.8885088	3.6558746	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
0.0067379	-0.7570750	[illegible]	[illegible]	-0.7111196	17.9646426	[illegible]	[illegible]	[illegible]	-0.2627046	[illegible]	[illegible]	-3.8906583	3.6591624	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

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 0.9931253 0.9931253 0.9931253 0.9920677 0.9910100
##  [8] 0.9899524 0.9899524 0.9910100 0.9915389 0.9920677 0.9931253 0.9952406
## [15] 0.9957694 0.9962983 0.9968271 0.9968271 0.9968271 0.9968271 0.9968271
## [22] 0.9968271 0.9962983 0.9962983 0.9957694 0.9957694 0.9957694 0.9952406
## [29] 0.9957694 0.9952406 0.9931253 0.9925965 0.9925965 0.9931253 0.9931253
## [36] 0.9936542 0.9931253 0.9920677 0.9920677 0.9915389 0.9894236 0.9894236
## [43] 0.9894236 0.9894236 0.9746166 0.9746166 0.9746166 0.9746166 0.9746166
## [50] 0.9746166
##                res       auc
##  [1,] 1.752945e+02 0.5000000
##  [2,] 1.424496e+02 0.5000000
##  [3,] 1.157589e+02 0.9931253
##  [4,] 9.406920e+01 0.9931253
##  [5,] 7.644348e+01 0.9931253
##  [6,] 6.212030e+01 0.9920677
##  [7,] 5.048084e+01 0.9910100
##  [8,] 4.102226e+01 0.9899524
##  [9,] 3.333594e+01 0.9899524
## [10,] 2.708979e+01 0.9910100
## [11,] 2.201399e+01 0.9915389
## [12,] 1.788924e+01 0.9920677
## [13,] 1.453734e+01 0.9931253
## [14,] 1.181348e+01 0.9952406
## [15,] 9.599994e+00 0.9957694
## [16,] 7.801246e+00 0.9962983
## [17,] 6.339530e+00 0.9968271
## [18,] 5.151694e+00 0.9968271
## [19,] 4.186423e+00 0.9968271
## [20,] 3.402015e+00 0.9968271
## [21,] 2.764580e+00 0.9968271
## [22,] 2.246582e+00 0.9968271
## [23,] 1.825641e+00 0.9962983
## [24,] 1.483571e+00 0.9962983
## [25,] 1.205595e+00 0.9957694
## [26,] 9.797031e-01 0.9957694
## [27,] 7.961365e-01 0.9957694
## [28,] 6.469647e-01 0.9952406
## [29,] 5.257431e-01 0.9957694
## [30,] 4.272348e-01 0.9952406
## [31,] 3.471840e-01 0.9931253
## [32,] 2.821323e-01 0.9925965
## [33,] 2.292692e-01 0.9925965
## [34,] 1.863111e-01 0.9931253
```

```
## [35,] 1.514021e-01 0.9931253
## [36,] 1.230339e-01 0.9936542
## [37,] 9.998112e-02 0.9931253
## [38,] 8.124769e-02 0.9920677
## [39,] 6.602434e-02 0.9920677
## [40,] 5.365338e-02 0.9915389
## [41,] 4.360037e-02 0.9894236
## [42,] 3.543098e-02 0.9894236
## [43,] 2.879229e-02 0.9894236
## [44,] 2.339749e-02 0.9894236
## [45,] 1.901351e-02 0.9746166
## [46,] 1.545096e-02 0.9746166
## [47,] 1.255592e-02 0.9746166
## [48,] 1.020332e-02 0.9746166
## [49,] 8.291527e-03 0.9746166
## [50,] 6.737947e-03 0.9746166
##  [1] 0.5000000 0.5000000 0.9456989 0.9532258 0.9715054 0.9768817 0.9790323
##  [8] 0.9795699 0.9806452 0.9860215 0.9876344 0.9897849 0.9919355 0.9930108
## [15] 0.9935484 0.9940860 0.9940860 0.9946237 0.9946237 0.9962366 0.9962366
## [22] 0.9962366 0.9962366 0.9962366 0.9962366 0.9978495 0.9978495 0.9973118
## [29] 0.9978495 0.9973118 0.9967742 0.9967742 0.9951613 0.9946237 0.9935484
## [36] 0.9935484 0.9935484 0.9935484 0.9919355 0.9924731 0.9919355 0.9919355
## [43] 0.9919355 0.9913978 0.9913978 0.9913978 0.9913978 0.9913978 0.9919355
## [50] 0.9919355
##                res       auc       auc
##  [1,] 1.752945e+02 0.5000000 0.5000000
##  [2,] 1.424496e+02 0.5000000 0.5000000
##  [3,] 1.157589e+02 0.9931253 0.9456989
##  [4,] 9.406920e+01 0.9931253 0.9532258
##  [5,] 7.644348e+01 0.9931253 0.9715054
##  [6,] 6.212030e+01 0.9920677 0.9768817
##  [7,] 5.048084e+01 0.9910100 0.9790323
##  [8,] 4.102226e+01 0.9899524 0.9795699
##  [9,] 3.333594e+01 0.9899524 0.9806452
## [10,] 2.708979e+01 0.9910100 0.9860215
## [11,] 2.201399e+01 0.9915389 0.9876344
## [12,] 1.788924e+01 0.9920677 0.9897849
## [13,] 1.453734e+01 0.9931253 0.9919355
## [14,] 1.181348e+01 0.9952406 0.9930108
## [15,] 9.599994e+00 0.9957694 0.9935484
## [16,] 7.801246e+00 0.9962983 0.9940860
## [17,] 6.339530e+00 0.9968271 0.9940860
## [18,] 5.151694e+00 0.9968271 0.9946237
## [19,] 4.186423e+00 0.9968271 0.9946237
## [20,] 3.402015e+00 0.9968271 0.9962366
## [21,] 2.764580e+00 0.9968271 0.9962366
## [22,] 2.246582e+00 0.9968271 0.9962366
## [23,] 1.825641e+00 0.9962983 0.9962366
## [24,] 1.483571e+00 0.9962983 0.9962366
## [25,] 1.205595e+00 0.9957694 0.9962366
## [26,] 9.797031e-01 0.9957694 0.9978495
## [27,] 7.961365e-01 0.9957694 0.9978495
## [28,] 6.469647e-01 0.9952406 0.9973118
## [29,] 5.257431e-01 0.9957694 0.9978495
```

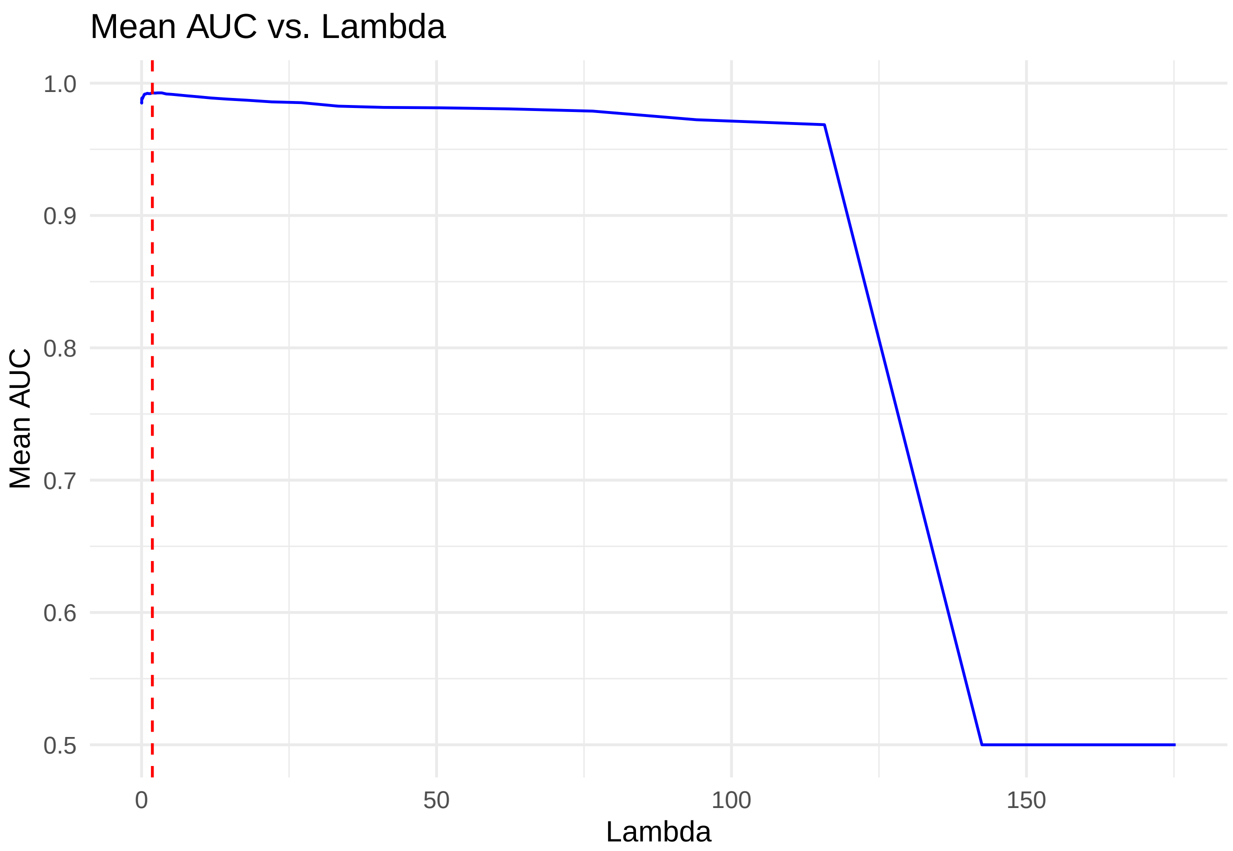
```
## [30,] 4.272348e-01 0.9952406 0.9973118
## [31,] 3.471840e-01 0.9931253 0.9967742
## [32,] 2.821323e-01 0.9925965 0.9967742
## [33,] 2.292692e-01 0.9925965 0.9951613
## [34,] 1.863111e-01 0.9931253 0.9946237
## [35,] 1.514021e-01 0.9931253 0.9935484
## [36,] 1.230339e-01 0.9936542 0.9935484
## [37,] 9.998112e-02 0.9931253 0.9935484
## [38,] 8.124769e-02 0.9920677 0.9935484
## [39,] 6.602434e-02 0.9920677 0.9919355
## [40,] 5.365338e-02 0.9915389 0.9924731
## [41,] 4.360037e-02 0.9894236 0.9919355
## [42,] 3.543098e-02 0.9894236 0.9919355
## [43,] 2.879229e-02 0.9894236 0.9919355
## [44,] 2.339749e-02 0.9894236 0.9913978
## [45,] 1.901351e-02 0.9746166 0.9913978
## [46,] 1.545096e-02 0.9746166 0.9913978
## [47,] 1.255592e-02 0.9746166 0.9913978
## [48,] 1.020332e-02 0.9746166 0.9913978
## [49,] 8.291527e-03 0.9746166 0.9919355
## [50,] 6.737947e-03 0.9746166 0.9919355
##  [1] 0.5000000 0.5000000 0.9474747 0.9540404 0.9585859 0.9590909 0.9631313
##  [8] 0.9636364 0.9641414 0.9681818 0.9702020 0.9742424 0.9757576 0.9767677
## [15] 0.9813131 0.9833333 0.9858586 0.9873737 0.9883838 0.9898990 0.9909091
## [22] 0.9904040 0.9898990 0.9868687 0.9858586 0.9848485 0.9843434 0.9833333
## [29] 0.9823232 0.9813131 0.9787879 0.9772727 0.9757576 0.9737374 0.9712121
## [36] 0.9696970 0.9681818 0.9666667 0.9676768 0.9676768 0.9671717 0.9671717
## [43] 0.9671717 0.9661616 0.9661616 0.9661616 0.9656566 0.9656566 0.9661616
## [50] 0.9661616
##                res       auc       auc       auc
##  [1,] 1.752945e+02 0.5000000 0.5000000 0.5000000
##  [2,] 1.424496e+02 0.5000000 0.5000000 0.5000000
##  [3,] 1.157589e+02 0.9931253 0.9456989 0.9474747
##  [4,] 9.406920e+01 0.9931253 0.9532258 0.9540404
##  [5,] 7.644348e+01 0.9931253 0.9715054 0.9585859
##  [6,] 6.212030e+01 0.9920677 0.9768817 0.9590909
##  [7,] 5.048084e+01 0.9910100 0.9790323 0.9631313
##  [8,] 4.102226e+01 0.9899524 0.9795699 0.9636364
##  [9,] 3.333594e+01 0.9899524 0.9806452 0.9641414
## [10,] 2.708979e+01 0.9910100 0.9860215 0.9681818
## [11,] 2.201399e+01 0.9915389 0.9876344 0.9702020
## [12,] 1.788924e+01 0.9920677 0.9897849 0.9742424
## [13,] 1.453734e+01 0.9931253 0.9919355 0.9757576
## [14,] 1.181348e+01 0.9952406 0.9930108 0.9767677
## [15,] 9.599994e+00 0.9957694 0.9935484 0.9813131
## [16,] 7.801246e+00 0.9962983 0.9940860 0.9833333
## [17,] 6.339530e+00 0.9968271 0.9940860 0.9858586
## [18,] 5.151694e+00 0.9968271 0.9946237 0.9873737
## [19,] 4.186423e+00 0.9968271 0.9946237 0.9883838
## [20,] 3.402015e+00 0.9968271 0.9962366 0.9898990
## [21,] 2.764580e+00 0.9968271 0.9962366 0.9909091
## [22,] 2.246582e+00 0.9968271 0.9962366 0.9904040
## [23,] 1.825641e+00 0.9962983 0.9962366 0.9898990
## [24,] 1.483571e+00 0.9962983 0.9962366 0.9868687
```

```
## [25,] 1.205595e+00 0.9957694 0.9962366 0.9858586
## [26,] 9.797031e-01 0.9957694 0.9978495 0.9848485
## [27,] 7.961365e-01 0.9957694 0.9978495 0.9843434
## [28,] 6.469647e-01 0.9952406 0.9973118 0.9833333
## [29,] 5.257431e-01 0.9957694 0.9978495 0.9823232
## [30,] 4.272348e-01 0.9952406 0.9973118 0.9813131
## [31,] 3.471840e-01 0.9931253 0.9967742 0.9787879
## [32,] 2.821323e-01 0.9925965 0.9967742 0.9772727
## [33,] 2.292692e-01 0.9925965 0.9951613 0.9757576
## [34,] 1.863111e-01 0.9931253 0.9946237 0.9737374
## [35,] 1.514021e-01 0.9931253 0.9935484 0.9712121
## [36,] 1.230339e-01 0.9936542 0.9935484 0.9696970
## [37,] 9.998112e-02 0.9931253 0.9935484 0.9681818
## [38,] 8.124769e-02 0.9920677 0.9935484 0.9666667
## [39,] 6.602434e-02 0.9920677 0.9919355 0.9676768
## [40,] 5.365338e-02 0.9915389 0.9924731 0.9676768
## [41,] 4.360037e-02 0.9894236 0.9919355 0.9671717
## [42,] 3.543098e-02 0.9894236 0.9919355 0.9671717
## [43,] 2.879229e-02 0.9894236 0.9919355 0.9671717
## [44,] 2.339749e-02 0.9894236 0.9913978 0.9661616
## [45,] 1.901351e-02 0.9746166 0.9913978 0.9661616
## [46,] 1.545096e-02 0.9746166 0.9913978 0.9661616
## [47,] 1.255592e-02 0.9746166 0.9913978 0.9656566
## [48,] 1.020332e-02 0.9746166 0.9913978 0.9656566
## [49,] 8.291527e-03 0.9746166 0.9919355 0.9661616
## [50,] 6.737947e-03 0.9746166 0.9919355 0.9661616
##  [1] 0.5000000 0.5000000 0.9700742 0.9708686 0.9772246 0.9788136 0.9777542
##  [8] 0.9782839 0.9804025 0.9830508 0.9819915 0.9814619 0.9804025 0.9804025
## [15] 0.9798729 0.9793432 0.9793432 0.9793432 0.9798729 0.9804025 0.9804025
## [22] 0.9804025 0.9825212 0.9830508 0.9846398 0.9856992 0.9867585 0.9878178
## [29] 0.9878178 0.9883475 0.9888771 0.9894068 0.9894068 0.9904661 0.9909958
## [36] 0.9904661 0.9909958 0.9899364 0.9899364 0.9920551 0.9931144 0.9931144
## [43] 0.9936441 0.9931144 0.9931144 0.9936441 0.9936441 0.9936441 0.9936441
## [50] 0.9936441
##                res       auc       auc       auc       auc
##  [1,] 1.752945e+02 0.5000000 0.5000000 0.5000000 0.5000000
##  [2,] 1.424496e+02 0.5000000 0.5000000 0.5000000 0.5000000
##  [3,] 1.157589e+02 0.9931253 0.9456989 0.9474747 0.9700742
##  [4,] 9.406920e+01 0.9931253 0.9532258 0.9540404 0.9708686
##  [5,] 7.644348e+01 0.9931253 0.9715054 0.9585859 0.9772246
##  [6,] 6.212030e+01 0.9920677 0.9768817 0.9590909 0.9788136
##  [7,] 5.048084e+01 0.9910100 0.9790323 0.9631313 0.9777542
##  [8,] 4.102226e+01 0.9899524 0.9795699 0.9636364 0.9782839
##  [9,] 3.333594e+01 0.9899524 0.9806452 0.9641414 0.9804025
## [10,] 2.708979e+01 0.9910100 0.9860215 0.9681818 0.9830508
## [11,] 2.201399e+01 0.9915389 0.9876344 0.9702020 0.9819915
## [12,] 1.788924e+01 0.9920677 0.9897849 0.9742424 0.9814619
## [13,] 1.453734e+01 0.9931253 0.9919355 0.9757576 0.9804025
## [14,] 1.181348e+01 0.9952406 0.9930108 0.9767677 0.9804025
## [15,] 9.599994e+00 0.9957694 0.9935484 0.9813131 0.9798729
## [16,] 7.801246e+00 0.9962983 0.9940860 0.9833333 0.9793432
## [17,] 6.339530e+00 0.9968271 0.9940860 0.9858586 0.9793432
## [18,] 5.151694e+00 0.9968271 0.9946237 0.9873737 0.9793432
## [19,] 4.186423e+00 0.9968271 0.9946237 0.9883838 0.9798729
```

```
## [20,] 3.402015e+00 0.9968271 0.9962366 0.9898990 0.9804025
## [21,] 2.764580e+00 0.9968271 0.9962366 0.9909091 0.9804025
## [22,] 2.246582e+00 0.9968271 0.9962366 0.9904040 0.9804025
## [23,] 1.825641e+00 0.9962983 0.9962366 0.9898990 0.9825212
## [24,] 1.483571e+00 0.9962983 0.9962366 0.9868687 0.9830508
## [25,] 1.205595e+00 0.9957694 0.9962366 0.9858586 0.9846398
## [26,] 9.797031e-01 0.9957694 0.9978495 0.9848485 0.9856992
## [27,] 7.961365e-01 0.9957694 0.9978495 0.9843434 0.9867585
## [28,] 6.469647e-01 0.9952406 0.9973118 0.9833333 0.9878178
## [29,] 5.257431e-01 0.9957694 0.9978495 0.9823232 0.9878178
## [30,] 4.272348e-01 0.9952406 0.9973118 0.9813131 0.9883475
## [31,] 3.471840e-01 0.9931253 0.9967742 0.9787879 0.9888771
## [32,] 2.821323e-01 0.9925965 0.9967742 0.9772727 0.9894068
## [33,] 2.292692e-01 0.9925965 0.9951613 0.9757576 0.9894068
## [34,] 1.863111e-01 0.9931253 0.9946237 0.9737374 0.9904661
## [35,] 1.514021e-01 0.9931253 0.9935484 0.9712121 0.9909958
## [36,] 1.230339e-01 0.9936542 0.9935484 0.9696970 0.9904661
## [37,] 9.998112e-02 0.9931253 0.9935484 0.9681818 0.9909958
## [38,] 8.124769e-02 0.9920677 0.9935484 0.9666667 0.9899364
## [39,] 6.602434e-02 0.9920677 0.9919355 0.9676768 0.9899364
## [40,] 5.365338e-02 0.9915389 0.9924731 0.9676768 0.9920551
## [41,] 4.360037e-02 0.9894236 0.9919355 0.9671717 0.9931144
## [42,] 3.543098e-02 0.9894236 0.9919355 0.9671717 0.9931144
## [43,] 2.879229e-02 0.9894236 0.9919355 0.9671717 0.9936441
## [44,] 2.339749e-02 0.9894236 0.9913978 0.9661616 0.9931144
## [45,] 1.901351e-02 0.9746166 0.9913978 0.9661616 0.9931144
## [46,] 1.545096e-02 0.9746166 0.9913978 0.9661616 0.9936441
## [47,] 1.255592e-02 0.9746166 0.9913978 0.9656566 0.9936441
## [48,] 1.020332e-02 0.9746166 0.9913978 0.9656566 0.9936441
## [49,] 8.291527e-03 0.9746166 0.9919355 0.9661616 0.9936441
## [50,] 6.737947e-03 0.9746166 0.9919355 0.9661616 0.9936441
##  [1] 0.5000000 0.5000000 0.9867367 0.9904905 0.9939940 0.9959960 0.9959960
##  [8] 0.9969970 0.9979980 0.9979980 0.9979980 0.9979980 0.9984985 0.9984985
## [15] 0.9979980 0.9989990 0.9989990 0.9994995 0.9994995 1.0000000 0.9989990
## [22] 0.9984985 0.9984985 0.9979980 0.9979980 0.9974975 0.9954955 0.9949950
## [29] 0.9949950 0.9944945 0.9939940 0.9934935 0.9929930 0.9929930 0.9929930
## [36] 0.9974975 0.9984985 0.9994995 0.9994995 0.9994995 0.9994995 0.9994995
## [43] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
## [50] 1.0000000
##                  res       auc       auc       auc       auc       auc
##  [1,] 1.752945e+02 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000
##  [2,] 1.424496e+02 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000
##  [3,] 1.157589e+02 0.9931253 0.9456989 0.9474747 0.9700742 0.9867367
##  [4,] 9.406920e+01 0.9931253 0.9532258 0.9540404 0.9708686 0.9904905
##  [5,] 7.644348e+01 0.9931253 0.9715054 0.9585859 0.9772246 0.9939940
##  [6,] 6.212030e+01 0.9920677 0.9768817 0.9590909 0.9788136 0.9959960
##  [7,] 5.048084e+01 0.9910100 0.9790323 0.9631313 0.9777542 0.9959960
##  [8,] 4.102226e+01 0.9899524 0.9795699 0.9636364 0.9782839 0.9969970
##  [9,] 3.333594e+01 0.9899524 0.9806452 0.9641414 0.9804025 0.9979980
## [10,] 2.708979e+01 0.9910100 0.9860215 0.9681818 0.9830508 0.9979980
## [11,] 2.201399e+01 0.9915389 0.9876344 0.9702020 0.9819915 0.9979980
## [12,] 1.788924e+01 0.9920677 0.9897849 0.9742424 0.9814619 0.9979980
## [13,] 1.453734e+01 0.9931253 0.9919355 0.9757576 0.9804025 0.9984985
## [14,] 1.181348e+01 0.9952406 0.9930108 0.9767677 0.9804025 0.9984985
```

```
## [15,] 9.599994e+00 0.9957694 0.9935484 0.9813131 0.9798729 0.9979980
## [16,] 7.801246e+00 0.9962983 0.9940860 0.9833333 0.9793432 0.9989990
## [17,] 6.339530e+00 0.9968271 0.9940860 0.9858586 0.9793432 0.9989990
## [18,] 5.151694e+00 0.9968271 0.9946237 0.9873737 0.9793432 0.9994995
## [19,] 4.186423e+00 0.9968271 0.9946237 0.9883838 0.9798729 0.9994995
## [20,] 3.402015e+00 0.9968271 0.9962366 0.9898990 0.9804025 1.0000000
## [21,] 2.764580e+00 0.9968271 0.9962366 0.9909091 0.9804025 0.9989990
## [22,] 2.246582e+00 0.9968271 0.9962366 0.9904040 0.9804025 0.9984985
## [23,] 1.825641e+00 0.9962983 0.9962366 0.9898990 0.9825212 0.9984985
## [24,] 1.483571e+00 0.9962983 0.9962366 0.9868687 0.9830508 0.9979980
## [25,] 1.205595e+00 0.9957694 0.9962366 0.9858586 0.9846398 0.9979980
## [26,] 9.797031e-01 0.9957694 0.9978495 0.9848485 0.9856992 0.9974975
## [27,] 7.961365e-01 0.9957694 0.9978495 0.9843434 0.9867585 0.9954955
## [28,] 6.469647e-01 0.9952406 0.9973118 0.9833333 0.9878178 0.9949950
## [29,] 5.257431e-01 0.9957694 0.9978495 0.9823232 0.9878178 0.9949950
## [30,] 4.272348e-01 0.9952406 0.9973118 0.9813131 0.9883475 0.9944945
## [31,] 3.471840e-01 0.9931253 0.9967742 0.9787879 0.9888771 0.9939940
## [32,] 2.821323e-01 0.9925965 0.9967742 0.9772727 0.9894068 0.9934935
## [33,] 2.292692e-01 0.9925965 0.9951613 0.9757576 0.9894068 0.9929930
## [34,] 1.863111e-01 0.9931253 0.9946237 0.9737374 0.9904661 0.9929930
## [35,] 1.514021e-01 0.9931253 0.9935484 0.9712121 0.9909958 0.9929930
## [36,] 1.230339e-01 0.9936542 0.9935484 0.9696970 0.9904661 0.9974975
## [37,] 9.998112e-02 0.9931253 0.9935484 0.9681818 0.9909958 0.9984985
## [38,] 8.124769e-02 0.9920677 0.9935484 0.9666667 0.9899364 0.9994995
## [39,] 6.602434e-02 0.9920677 0.9919355 0.9676768 0.9899364 0.9994995
## [40,] 5.365338e-02 0.9915389 0.9924731 0.9676768 0.9920551 0.9994995
## [41,] 4.360037e-02 0.9894236 0.9919355 0.9671717 0.9931144 0.9994995
## [42,] 3.543098e-02 0.9894236 0.9919355 0.9671717 0.9931144 0.9994995
## [43,] 2.879229e-02 0.9894236 0.9919355 0.9671717 0.9936441 1.0000000
## [44,] 2.339749e-02 0.9894236 0.9913978 0.9661616 0.9931144 1.0000000
## [45,] 1.901351e-02 0.9746166 0.9913978 0.9661616 0.9931144 1.0000000
## [46,] 1.545096e-02 0.9746166 0.9913978 0.9661616 0.9936441 1.0000000
## [47,] 1.255592e-02 0.9746166 0.9913978 0.9656566 0.9936441 1.0000000
## [48,] 1.020332e-02 0.9746166 0.9913978 0.9656566 0.9936441 1.0000000
## [49,] 8.291527e-03 0.9746166 0.9919355 0.9661616 0.9936441 1.0000000
## [50,] 6.737947e-03 0.9746166 0.9919355 0.9661616 0.9936441 1.0000000
```

```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: 0.9958
```

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

auc_full	auc_lasso
0.9771242	0.9957516

```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)	-85.841618693509	-0.631891749918733
radius_mean	0.482809216338564	2.07950517324475
texture_mean	1.22782799498286	1.21495895057423
smoothness_mean	-61.2848250472609	0
concavity_mean	277.147953900727	0.915894304171588
symmetry_mean	-174.873046711938	0
fractal_dimension_mean	-268.503538734735	0
radius_se	68.8555271776555	2.23661374208485
texture_se	-0.498284713431817	0
smoothness_se	1374.2230814889	0
compactness_se	42.7436942672398	-0.527415419852414
concavity_se	-150.677448596641	0
concave_points_se	731.297840302427	0
symmetry_se	-1308.68956822617	-0.385186799354295
fractal_dimension_se	-6434.27652769617	-0.421464230008035
smoothness_worst	-133.987895735445	0.464232406689049
compactness_worst	-104.622145944305	0
concave_points_worst	161.993778687915	2.1144068366775
symmetry_worst	180.871288833035	0.745540291717346
fractal_dimension_worst	807.843869038758	0