STAT-627 Project

Danny Tapp

Question 2:

How well do the variables age, sex, race/ethnicity, income, and education level predict the percentage of adults who achieve at least 150 minutes a week of moderate-intensity aerobic physical activity across US states?

```r
library(tidyverse)

### Load in data
food <- read_csv("food.csv")
```

```r
### Get the question about 150+ min of exercise a week and the necessary variables
df_model <- food |>
  filter(
    TopicID == "PA1",
    QuestionID == "Q043",
    !is.na(Data_Value),
    StratificationCategory1 %in% c("Age (years)", "Sex", "Race/Ethnicity", "Income", "Educ
  )

### Pivot wider so each stratum becomes it own column
df_wide <- df_model |>
  dplyr::select(LocationAbbr, YearStart,
         StratificationCategory1, Stratification1, Data_Value) |>
  unite(var, StratificationCategory1, Stratification1, sep = "_") |>
  pivot_wider(names_from = var, values_from = Data_Value)
```

```r
### Impute missing values using the median of each column
df_imp <- df_wide |>
  mutate(across(where(is.numeric), ~ replace_na(.x, median(.x, na.rm = TRUE))))
```

```r
### Extract the overall percentage values
df_overall <- food |>
  filter(
    TopicID == "PA1",
    QuestionID == "Q043",
    is.na(StratificationCategory1) |
      StratificationCategory1 %in% c("Overall", "Total") |
      Stratification1 %in% c("Overall", "Total", "OVR")
  ) |>
  dplyr::select(LocationAbbr, YearStart, PercentOverall = Data_Value)

### Join the two data sets
df_model2 <- df_imp |>
  left_join(df_overall, by = c("LocationAbbr", "YearStart"))
```

```r
### Predictor Matrix
X <- df_model2 |>
  dplyr::select(-LocationAbbr, -YearStart, -PercentOverall) |>
  as.matrix()

### Response
y <- df_model2$PercentOverall
```

```r
library(glmnet)
```

```
Loading required package: Matrix


Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack

Loaded glmnet 4.1-10
```

```r
### Standardize predictor matrix for ridge
X <- scale(X)
```

```r
set.seed(123)

### 10-fold cross-validation ridge regression
cv_ridge <- cv.glmnet(
  X, y,
  alpha = 0,
  nfolds = 10
)

### Lambda that minimizes cross-validation error
best_lambda <- cv_ridge$lambda.min

best_lambda
```

```
[1] 0.6391113
```

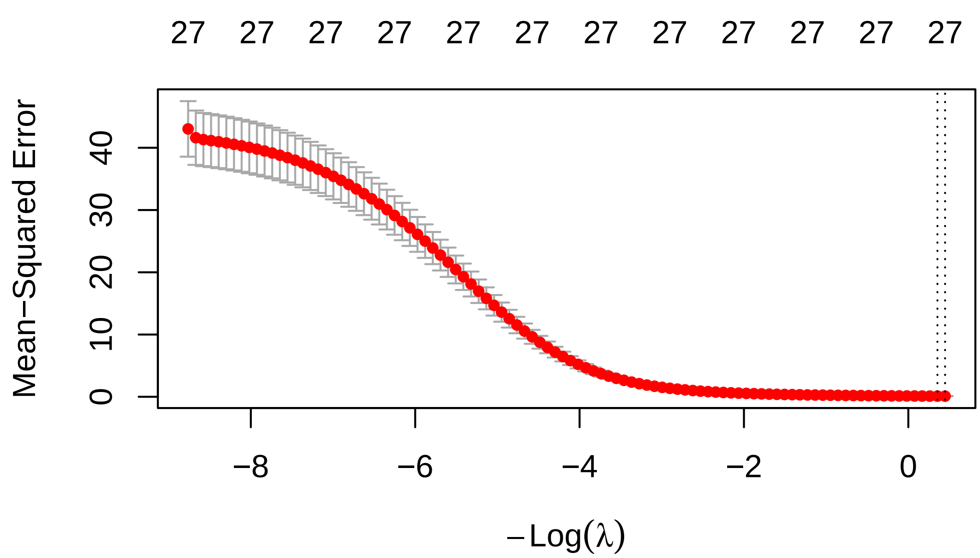
```r
### Fit ridge regression model using best lambda
ridge_final <- glmnet(
  X, y,
  alpha = 0,
  lambda = best_lambda
)

coef(ridge_final)
```

```
28 x 1 sparse Matrix of class "dgCMatrix"
                                                  s0
(Intercept)                             52.313836478
Income_$15,000 - $24,999                 0.082570466
Income_$25,000 - $34,999                 0.037845404
Income_$35,000 - $49,999                 0.117056342
Income_$50,000 - $74,999                 0.159823257
Income_$75,000 or greater                0.280781509
Age (years)_18 - 24                      0.290150181
Race/Ethnicity_2 or more races           0.015693338
Age (years)_25 - 34                      0.504075021
Age (years)_35 - 44                      0.534949137
Age (years)_45 - 54                      0.669258522
Age (years)_55 - 64                      0.724984474
Age (years)_65 or older                  0.641806827
```

```
Race/Ethnicity_American Indian/Alaska Native -0.031537102
Education_College graduate                    0.446487837
Income_Data not reported                      0.312152639
Sex_Female                                    0.686421573
Education_High school graduate                0.327526534
Race/Ethnicity_Hispanic                      -0.004699734
Income_Less than $15,000                      0.082084394
Education_Less than high school               0.149554801
Sex_Male                                      0.512572179
Race/Ethnicity_Non-Hispanic Black             0.003946892
Race/Ethnicity_Non-Hispanic White             0.162085726
Education_Some college or technical school    0.303347341
Race/Ethnicity_Other                         -0.012596834
Race/Ethnicity_Asian                         -0.006700693
Race/Ethnicity_Hawaiian/Pacific Islander      0.025866220
```

```
plot(cv_ridge)
```




```
### Mean CV error for each lambda
cv_ridge$cvm
```

```
 [1] 43.03147921 41.61640129 41.31845795 41.15142142 40.96928446 40.77080540
 [7] 40.55466370 40.31946070 40.06372089 39.78589457 39.48436235 39.15744176
[13] 38.80339634 38.42044758 38.00677926 37.56034756 37.07974591 36.56306982
[19] 36.00861001 35.41478977 34.78018375 34.10356408 33.38395063 32.62066481
[25] 31.81338585 30.96220808 30.06769747 29.13094518 28.15361554 27.13798555
[31] 26.08697286 25.00414913 23.89372886 22.76043422 21.60979309 20.44769241
[37] 19.28045962 18.11472645 16.95729698 15.81500470 14.69456192 13.60240772
[43] 12.54456112 11.52648559 10.55297068  9.62803545  8.75485708  7.93572708
[49]  7.17203622  6.46428876  5.81214572  5.21427544  4.66419219  4.16965332
[55]  3.72414897  3.32399594  2.96586103  2.64627074  2.36174634  2.10903413
[61]  1.88622613  1.69016382  1.51647600  1.36278307  1.22704428  1.10739536
[67]  1.00209975  0.90947168  0.82789297  0.75587360  0.69209089  0.63539743
[73]  0.58481110  0.53949728  0.49874927  0.46196964  0.42865374  0.39837534
[79]  0.37077411  0.34554477  0.32242758  0.30120022  0.28189063  0.26399008
[85]  0.24725793  0.23182344  0.21756529  0.20459397  0.19236128  0.18089199
[91]  0.17043307  0.16025013  0.15107611  0.14244222  0.13410294  0.12652571
[97]  0.11957724  0.11270228  0.10664839  0.09982398
```

```r
### Lambda with lowest CV error
cv_ridge$lambda.min
```

```
[1] 0.6391113
```

```r
### Lambda within 1 se of min
cv_ridge$lambda.1se
```

```
[1] 0.7014238
```

```r
### Best lambda's MSE
best_lambda <- cv_ridge$lambda.min
best_mse <- cv_ridge$cvm[cv_ridge$lambda == best_lambda]
```

```r
round(best_lambda,4)
```

```
[1] 0.6391
```

```r
round(best_mse,4)
```

```
[1] 0.0998
```

```r
sqrt(best_mse)
```

```
[1] 0.3159493
```

```r
### Take model coefficients and put them into a vector with their names
coef_vec <- as.numeric(coef(ridge_final))
names_vec <- rownames(coef(ridge_final))

### DF of coefficients
coef_df <- data.frame(
  variable = names_vec,
  coefficient = round(coef_vec,4),
  row.names = NULL
)

### Remove intercept
coef_df <- coef_df |>
  dplyr::filter(variable != "(Intercept)")

### Top 10 predictors based on largest aboslute coefficients
top10 <- coef_df |>
  dplyr::arrange(desc(abs(coefficient))) |>
  dplyr::slice(1:10)

top10
```

```
                        variable coefficient
1            Age (years)_55 - 64      0.7250
2                     Sex_Female      0.6864
3            Age (years)_45 - 54      0.6693
4        Age (years)_65 or older      0.6418
5            Age (years)_35 - 44      0.5349
6                       Sex_Male      0.5126
7            Age (years)_25 - 34      0.5041
8      Education_College graduate      0.4465
9  Education_High school graduate      0.3275
10       Income_Data not reported      0.3122
```

```r
### Age 18 - 24 has the smallest SD
### Other age demographics have larger between state variance
df_wide |>
  summarize(across(starts_with("Age"), ~ sd(.x, na.rm = TRUE)))
```

```
# A tibble: 1 x 6
  `Age (years)_18 - 24` `Age (years)_25 - 34` `Age (years)_35 - 44`
                  <dbl>                 <dbl>                 <dbl>
1                  6.10                  6.61                  7.12
# i 3 more variables: `Age (years)_45 - 54` <dbl>, `Age (years)_55 - 64` <dbl>,
#   `Age (years)_65 or older` <dbl>
```

```r
### Create Lasso for comparison
cv_lasso <- cv.glmnet(X, y, alpha = 1, nfolds = 10)
best_lambda_lasso <- cv_lasso$lambda.min

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

```
28 x 1 sparse Matrix of class "dgCMatrix"
                                                       s0
(Intercept)                                  5.231384e+01
Income_$15,000 - $24,999                     .
Income_$25,000 - $34,999                     .
Income_$35,000 - $49,999                     .
Income_$50,000 - $74,999                     .
Income_$75,000 or greater                    .
Age (years)_18 - 24                          .
Race/Ethnicity_2 or more races               .
Age (years)_25 - 34                          .
Age (years)_35 - 44                          .
Age (years)_45 - 54                          .
Age (years)_55 - 64                          .
Age (years)_65 or older                      .
Race/Ethnicity_American Indian/Alaska Native .
Education_College graduate                   .
Income_Data not reported                     .
Sex_Female                                   3.468190e+00
Education_High school graduate               6.928607e-04
Race/Ethnicity_Hispanic                      .
```

```
Income_Less than $15,000                         .
Education_Less than high school                  .
Sex_Male                                         3.067647e+00
Race/Ethnicity_Non-Hispanic Black                .
Race/Ethnicity_Non-Hispanic White                .
Education_Some college or technical school       .
Race/Ethnicity_Other                             .
Race/Ethnicity_Asian                             .
Race/Ethnicity_Hawaiian/Pacific Islander         .
```

```r
### Ridge coefficients
ridge_coef <- coef(ridge_final)
ridge_df <- data.frame(
  variable = ridge_coef@Dimnames[[1]],
  ridge = as.numeric(ridge_coef)
) |>
  dplyr::filter(variable != "(Intercept)")
```

```r
### Lasso coefficients
lasso_coef <- coef(lasso_final)
lasso_df <- data.frame(
  variable = lasso_coef@Dimnames[[1]],
  lasso = as.numeric(lasso_coef)
) |>
  dplyr::filter(variable != "(Intercept)")
```

```r
### Comparison
coef_compare <- ridge_df |>
  full_join(lasso_df, by = "variable") |>
  arrange(desc(abs(ridge)))

coef_compare
```

```
                              variable       ridge        lasso
1                   Age (years)_55 - 64  0.724984474 0.0000000000
2                            Sex_Female  0.686421573 3.4681900251
3                   Age (years)_45 - 54  0.669258522 0.0000000000
4                Age (years)_65 or older 0.641806827 0.0000000000
5                   Age (years)_35 - 44  0.534949137 0.0000000000
6                              Sex_Male  0.512572179 3.0676473866
7                   Age (years)_25 - 34  0.504075021 0.0000000000
```

```
8                    Education_College graduate  0.446487837 0.0000000000
9                Education_High school graduate  0.327526534 0.0006928607
10                     Income_Data not reported  0.312152639 0.0000000000
11   Education_Some college or technical school  0.303347341 0.0000000000
12                          Age (years)_18 - 24  0.290150181 0.0000000000
13                     Income_$75,000 or greater  0.280781509 0.0000000000
14           Race/Ethnicity_Non-Hispanic White  0.162085726 0.0000000000
15                    Income_$50,000 - $74,999  0.159823257 0.0000000000
16             Education_Less than high school  0.149554801 0.0000000000
17                    Income_$35,000 - $49,999  0.117056342 0.0000000000
18                    Income_$15,000 - $24,999  0.082570466 0.0000000000
19                    Income_Less than $15,000  0.082084394 0.0000000000
20                    Income_$25,000 - $34,999  0.037845404 0.0000000000
21 Race/Ethnicity_American Indian/Alaska Native -0.031537102 0.0000000000
22     Race/Ethnicity_Hawaiian/Pacific Islander  0.025866220 0.0000000000
23               Race/Ethnicity_2 or more races  0.015693338 0.0000000000
24                         Race/Ethnicity_Other -0.012596834 0.0000000000
25                         Race/Ethnicity_Asian -0.006700693 0.0000000000
26                      Race/Ethnicity_Hispanic -0.004699734 0.0000000000
27            Race/Ethnicity_Non-Hispanic Black  0.003946892 0.0000000000
```

```r
coef_compare |> filter(lasso != 0) |> arrange(desc(abs(lasso)))
```

```
                        variable     ridge        lasso
1                     Sex_Female 0.6864216 3.4681900251
2                       Sex_Male 0.5125722 3.0676473866
3 Education_High school graduate 0.3275265 0.0006928607
```

```r
coef_table <- coef_compare |>
  mutate(
    ridge = round(ridge, 3),
    lasso = round(lasso, 3),
    selected_lasso = ifelse(lasso != 0, "Yes", "No")
  ) |>
  arrange(desc(abs(ridge)))

coef_table
```

```
                                       variable  ridge lasso selected_lasso
1                           Age (years)_55 - 64  0.725 0.000             No
```

```
2                                      Sex_Female  0.686 3.468           Yes
3                                Age (years)_45 - 54  0.669 0.000            No
4                            Age (years)_65 or older  0.642 0.000            No
5                                Age (years)_35 - 44  0.535 0.000            No
6                                        Sex_Male  0.513 3.068           Yes
7                                Age (years)_25 - 34  0.504 0.000            No
8                       Education_College graduate  0.446 0.000            No
9                   Education_High school graduate  0.328 0.001           Yes
10                        Income_Data not reported  0.312 0.000            No
11   Education_Some college or technical school  0.303 0.000            No
12                               Age (years)_18 - 24  0.290 0.000            No
13                         Income_$75,000 or greater  0.281 0.000            No
14            Race/Ethnicity_Non-Hispanic White  0.162 0.000            No
15                        Income_$50,000 - $74,999  0.160 0.000            No
16                Education_Less than high school  0.150 0.000            No
17                        Income_$35,000 - $49,999  0.117 0.000            No
18                        Income_$15,000 - $24,999  0.083 0.000            No
19                        Income_Less than $15,000  0.082 0.000            No
20                        Income_$25,000 - $34,999  0.038 0.000            No
21 Race/Ethnicity_American Indian/Alaska Native -0.032 0.000            No
22     Race/Ethnicity_Hawaiian/Pacific Islander  0.026 0.000            No
23                  Race/Ethnicity_2 or more races  0.016 0.000            No
24                            Race/Ethnicity_Other -0.013 0.000            No
25                            Race/Ethnicity_Asian -0.007 0.000            No
26                         Race/Ethnicity_Hispanic -0.005 0.000            No
27            Race/Ethnicity_Non-Hispanic Black  0.004 0.000            No
```

```r
cv_lasso <- cv.glmnet(X, y, alpha = 1, nfolds = 10)
best_lambda_lasso <- cv_lasso$lambda.min
best_lambda_lasso
```

```
[1] 0.1863043
```

```r
cv_lasso$cvm
```

```
 [1] 42.73349691 36.61296466 30.42158077 25.25675291 20.96938422 17.40971096
 [7] 14.45449206 12.00102406  9.96394613  8.27294849  6.86873971  5.70333076
[13]  4.73566200  3.93206103  3.26508905  2.71120368  2.25158205  1.86973849
[19]  1.55289711  1.28986611  1.07139185  0.89004873  0.73943940  0.61448261
[25]  0.51070507  0.42453705  0.35301347  0.29363518  0.24434502  0.20338786
```

```
[31]  0.16939427  0.14117923  0.11775730  0.09831155  0.08214916  0.06876364
[37]  0.05763779  0.04840952  0.04099392
```

```r
cv_lasso$lambda
```

```
 [1] 6.3911129 5.8233440 5.3060142 4.8346425 4.4051462 4.0138051 3.6572298
 [8] 3.3323315 3.0362964 2.7665602 2.5207866 2.2968469 2.0928013 1.9068826
[15] 1.7374804 1.5831275 1.4424868 1.3143402 1.1975779 1.0911883 0.9942502
[22] 0.9059237 0.8254440 0.7521138 0.6852981 0.6244181 0.5689465 0.5184029
[29] 0.4723494 0.4303872 0.3921528 0.3573150 0.3255721 0.2966492 0.2702957
[36] 0.2462834 0.2244042 0.2044688 0.1863043
```

```r
best_mse_lasso <- cv_lasso$cvm[cv_lasso$lambda == best_lambda_lasso]
best_mse_lasso
```

```
[1] 0.04099392
```

```r
lambda_1se_lasso <- cv_lasso$lambda.1se
lambda_1se_lasso
```

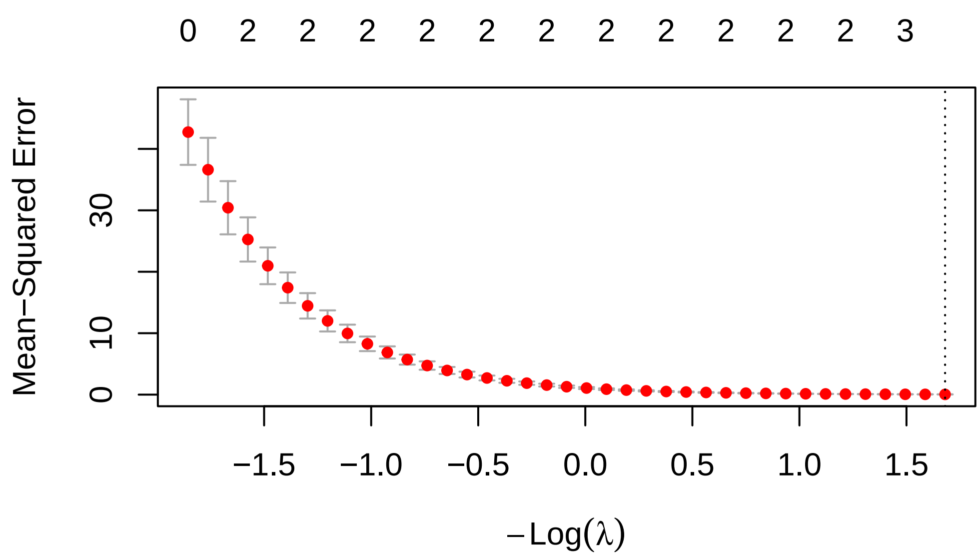
```
[1] 0.1863043
```

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

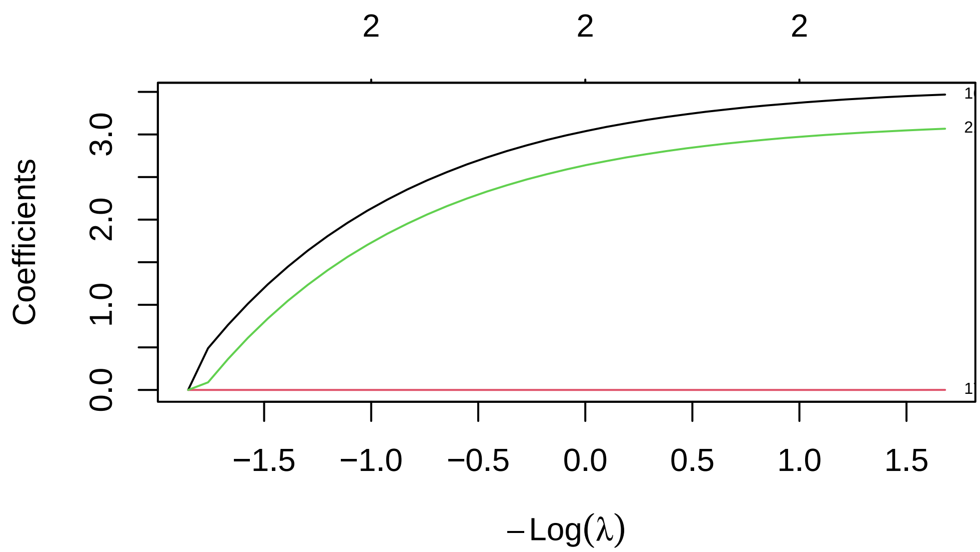
```
28 x 1 sparse Matrix of class "dgCMatrix"
                                                     s0
(Intercept)                                5.231384e+01
Income_$15,000 - $24,999                   .
Income_$25,000 - $34,999                   .
Income_$35,000 - $49,999                   .
Income_$50,000 - $74,999                   .
Income_$75,000 or greater                  .
Age (years)_18 - 24                        .
Race/Ethnicity_2 or more races             .
Age (years)_25 - 34                        .
Age (years)_35 - 44                        .
```

```
Age (years)_45 - 54                           .
Age (years)_55 - 64                           .
Age (years)_65 or older                       .
Race/Ethnicity_American Indian/Alaska Native .
Education_College graduate                    .
Income_Data not reported                      .
Sex_Female                                    3.468190e+00
Education_High school graduate                6.928607e-04
Race/Ethnicity_Hispanic                       .
Income_Less than $15,000                      .
Education_Less than high school               .
Sex_Male                                      3.067647e+00
Race/Ethnicity_Non-Hispanic Black             .
Race/Ethnicity_Non-Hispanic White             .
Education_Some college or technical school    .
Race/Ethnicity_Other                          .
Race/Ethnicity_Asian                          .
Race/Ethnicity_Hawaiian/Pacific Islander      .
```

```
plot(cv_lasso)
```



```r
plot(cv_lasso$glmnet.fit, xvar = "lambda", label = TRUE)
```



Question 5:

Can the percentage of adults in a state who engage in healthy behaviors (exercising and eating fruits/vegetables) be used to classify whether a state's obesity rate is above or below the national median?

```r
### Get questions for fruit intake, vegetable intake, and obesity
### Get overall values
veggie_model <- food |>
  filter(
    QuestionID %in% c("Q018", "Q019", "Q036"),
    StratificationCategoryId1 == "OVR",
    !is.na(Data_Value)
  )
```

```r
### Rename to make it easier to distinguish variables
veggie_clean <- veggie_model |>
  mutate(
```

```r
    Var = case_when(
      QuestionID == "Q018" ~ "FruitUnhealthy",
      QuestionID == "Q019" ~ "VegUnhealthy",
      QuestionID == "Q036" ~ "Obesity"
    )
  )
```

```r
### Pivot wider so each row is a state-year with all 3 variables
veggie_wide <- veggie_clean |>
  dplyr::select(LocationAbbr, YearStart, Var, Data_Value) |>
  pivot_wider(
    names_from = Var,
    values_from = Data_Value
  )
```

```r
### Change to percent healthy
veggie_wide <- veggie_wide |>
  mutate(
    FruitHealthy = 100 - FruitUnhealthy,
    VegHealthy = 100 - VegUnhealthy
  )
```

```r
### Get national median obesity rate
national_median <- veggie_wide |>
  filter(LocationAbbr == "US") |>
  summarize(med = median(Obesity, na.rm = TRUE)) |>
  pull(med)
```

```r
### Classify which states are above or below the national median obesity
veggie_wide <- veggie_wide |>
  mutate(
    ObesityClass = ifelse(
      Obesity > national_median,
      "High",
      "Low"
    )
  )
```

```r
### Remove US overall
veggie_model_ready <- veggie_wide |>
  filter(!LocationAbbr == "US")

veggie_lqda <- veggie_model_ready |>
  dplyr::select(LocationAbbr:ObesityClass) |>
  drop_na()
```

```r
library(MASS)
```

```

Attaching package: 'MASS'


The following object is masked from 'package:dplyr':

    select
```

```r
### Fit LDA model
lda_fit <- lda(
  ObesityClass ~ FruitHealthy + VegHealthy,
  data = veggie_lqda
)

lda_fit
```

```
Call:
lda(ObesityClass ~ FruitHealthy + VegHealthy, data = veggie_lqda)

Prior probabilities of groups:
     High       Low
0.6772152 0.3227848

Group means:
     FruitHealthy VegHealthy
High     58.60000   79.14299
Low      64.78039   81.69608

Coefficients of linear discriminants:
                     LD1
```

```
FruitHealthy  0.29609364
VegHealthy   -0.07452308
```

```r
### LDA coefficients
lda_fit$scaling
```

```
                     LD1
FruitHealthy  0.29609364
VegHealthy   -0.07452308
```

```r
### Look at averages between classes
veggie_lqda |>
  group_by(ObesityClass) |>
  summarise(
    n = n(),
    FruitHealthy = mean(FruitHealthy, na.rm = TRUE),
    VegHealthy   = mean(VegHealthy, na.rm = TRUE),
    Obesity      = mean(Obesity, na.rm = TRUE)
  )
```

```
# A tibble: 2 x 5
  ObesityClass     n FruitHealthy VegHealthy Obesity
  <chr>        <int>        <dbl>      <dbl>   <dbl>
1 High           107         58.6       79.1    34.4
2 Low             51         64.8       81.7    27.4
```

```r
### Priors
lda_fit$prior
```

```
     High       Low 
0.6772152 0.3227848 
```

```r
### Group means for each predictor by class
lda_fit$means
```

```
     FruitHealthy VegHealthy
High     58.60000   79.14299
Low      64.78039   81.69608
```

```r
### Fit QDA model
qda_fit <- qda(
  ObesityClass ~ FruitHealthy + VegHealthy,
  data = veggie_lqda
)

qda_fit
```

```
Call:
qda(ObesityClass ~ FruitHealthy + VegHealthy, data = veggie_lqda)

Prior probabilities of groups:
     High       Low 
0.6772152 0.3227848 

Group means:
     FruitHealthy VegHealthy
High     58.60000   79.14299
Low      64.78039   81.69608
```

```r
### Priors
qda_fit$prior
```

```
     High       Low 
0.6772152 0.3227848 
```

```r
### ### Group means for each predictor by class
qda_fit$means
```

```
     FruitHealthy VegHealthy
High     58.60000   79.14299
Low      64.78039   81.69608
```

```r
### Predict both LDA and QDA
lda_pred <- predict(lda_fit)
qda_pred <- predict(qda_fit)
```

```r
### LDA Confusion matrix
table(veggie_lqda$ObesityClass, lda_pred$class)
```

```
      High Low
High    94  13
Low     15  36
```

```r
### Classification rate
mean(lda_pred$class == veggie_lqda$ObesityClass)
```

```
[1] 0.8227848
```

```r
lda_pred <- predict(lda_fit)
lda_pred$posterior
```

```
          High         Low
1   0.98156447 0.018435533
2   0.53182265 0.468177348
3   0.51328736 0.486712641
4   0.97972267 0.020277328
5   0.09095388 0.909046120
6   0.17214454 0.827855458
7   0.09641853 0.903581471
8   0.40592294 0.594077063
9   0.09249060 0.907509400
10  0.24320540 0.756794601
11  0.74322586 0.256774142
12  0.48291193 0.517088067
13  0.44472916 0.555270840
14  0.12994815 0.870051847
15  0.87522336 0.124776635
16  0.36810469 0.631895309
17  0.66263350 0.337366498
18  0.96037264 0.039627356
19  0.96956681 0.030433191
20  0.14977901 0.850220988
21  0.25935562 0.740644376
22  0.08636245 0.913637554
```

23	0.48645508	0.513544919
24	0.12092737	0.879072634
25	0.98784718	0.012152821
26	0.86301954	0.136980463
27	0.58934891	0.410651087
28	0.50131665	0.498683353
29	0.52211305	0.477886946
30	0.08096931	0.919030688
31	0.19179600	0.808204004
32	0.54659502	0.453404981
33	0.15681503	0.843184966
34	0.68690391	0.313096086
35	0.49528891	0.504711094
36	0.63472194	0.365278056
37	0.99081228	0.009187725
38	0.19026213	0.809737865
39	0.28613114	0.713868859
40	0.21555732	0.784442683
41	0.88554674	0.114453265
42	0.59485560	0.405144405
43	0.81600426	0.183995739
44	0.72846797	0.271532030
45	0.11970343	0.880296574
46	0.06768558	0.932314425
47	0.59214472	0.407855280
48	0.12629267	0.873707332
49	0.98365035	0.016349650
50	0.11331453	0.886685473
51	0.72471979	0.275280208
52	0.93596863	0.064031375
53	0.99776520	0.002234803
54	0.98509275	0.014907246
55	0.95220958	0.047790423
56	0.63411387	0.365886130
57	0.98868928	0.011310722
58	0.30764453	0.692355473
59	0.52869731	0.471302687
60	0.20719185	0.792808147
61	0.50428952	0.495710478
62	0.45060991	0.549390090
63	0.62857831	0.371421691
64	0.95841783	0.041582173
65	0.80401835	0.195981646

66	0.59198641	0.408013586
67	0.56697269	0.433027309
68	0.89362340	0.106376596
69	0.80583415	0.194165848
70	0.89935265	0.100647346
71	0.98741949	0.012580507
72	0.98689668	0.013103324
73	0.52407519	0.475924810
74	0.51015225	0.489847745
75	0.18482649	0.815173513
76	0.81356675	0.186433254
77	0.29040565	0.709594347
78	0.99181539	0.008184607
79	0.93706529	0.062934710
80	0.91101625	0.088983749
81	0.76311434	0.236885665
82	0.92592037	0.074079631
83	0.47787502	0.522124985
84	0.92846127	0.071538730
85	0.21915719	0.780842810
86	0.71955300	0.280447001
87	0.82419531	0.175804693
88	0.94250122	0.057498777
89	0.99552745	0.004472551
90	0.47779325	0.522206754
91	0.63732315	0.362676847
92	0.38782653	0.612173472
93	0.96168413	0.038315867
94	0.80839972	0.191600283
95	0.95643928	0.043560720
96	0.85201251	0.147987486
97	0.42574304	0.574256956
98	0.18517237	0.814827628
99	0.71942071	0.280579287
100	0.44710054	0.552899458
101	0.98998376	0.010016240
102	0.65342599	0.346574007
103	0.93988701	0.060112994
104	0.99346833	0.006531674
105	0.99321348	0.006786518
106	0.98663315	0.013366850
107	0.94450749	0.055492508
108	0.88888679	0.111113206

109	0.98882874	0.011171264
110	0.34522852	0.654771480
111	0.69669595	0.303304053
112	0.37923684	0.620763162
113	0.78643868	0.213561319
114	0.19395335	0.806046654
115	0.90680743	0.093192569
116	0.92693196	0.073068044
117	0.94524635	0.054753650
118	0.82308678	0.176913217
119	0.95267401	0.047325989
120	0.86669118	0.133308818
121	0.96588737	0.034112626
122	0.98133604	0.018663963
123	0.99323491	0.006765095
124	0.48149743	0.518502568
125	0.54986499	0.450135006
126	0.56225394	0.437746057
127	0.82275249	0.177247513
128	0.42291822	0.577081779
129	0.98867461	0.011325391
130	0.96165997	0.038340026
131	0.86865856	0.131341438
132	0.93839661	0.061603391
133	0.88253765	0.117462352
134	0.53855430	0.461445701
135	0.30799366	0.692006340
136	0.85959127	0.140408727
137	0.40800204	0.591997964
138	0.87399108	0.126008923
139	0.95731249	0.042687509
140	0.95273308	0.047266921
141	0.99649035	0.003509650
142	0.69711129	0.302888706
143	0.78454541	0.215454589
144	0.34574727	0.654252735
145	0.93330324	0.066696759
146	0.96298895	0.037011050
147	0.98897364	0.011026358
148	0.80177339	0.198226612
149	0.59716658	0.402833421
150	0.38047204	0.619527964
151	0.82466966	0.175330336

```
152 0.43213072 0.567869277
153 0.98962375 0.010376253
154 0.71412678 0.285873216
155 0.96121910 0.038780896
156 0.84191785 0.158082150
157 0.97464403 0.025355967
158 0.46851523 0.531484766
```

```r
### QDA Confusion matrix
table(veggie_lqda$ObesityClass, qda_pred$class)
```

```
      High Low
 High   86  21
 Low     9  42
```

```r
### Classification Rate
mean(qda_pred$class == veggie_lqda$ObesityClass)
```

```
[1] 0.8101266
```

```r
### Grid for data points
grid <- expand.grid(
  FruitHealthy = seq(min(veggie_lqda$FruitHealthy), max(veggie_lqda$FruitHealthy), length.
  VegHealthy   = seq(min(veggie_lqda$VegHealthy),   max(veggie_lqda$VegHealthy),   length.
)

# Get LDA posterior for each grid point
grid$pred <- predict(lda_fit, newdata = grid)$class

### Plot
ggplot() +
  geom_tile(data = grid, aes(FruitHealthy, VegHealthy, fill = pred),
            alpha = 0.25) +
  geom_point(data = veggie_lqda,
             aes(FruitHealthy, VegHealthy, color = ObesityClass), size = 3) +
  labs(title = "LDA Classification Regions",
       subtitle = "Shaded regions are predicted class",
       fill = "Predicted Class") +
```

```
theme_minimal()
```