Homework 3: Spotify

1 Directions

1.1 Hot Tips

- To make HW go more smoothly, deepen your learning, & maximize happiness:
  - *Before* starting HW, *review* the last couple weeks of material.
  - Start early! HW isn't designed to plow through in 1 sitting.
- Whether currently working solo or with friends, reach out on Slack to invite others.
- WHEN you have questions:
  - Stop by office hours. Please remember that OH are for *group* discussion and exploring concepts / specific questions, not doing HW step by step.
  - Ask questions in the `#homework` channel on Slack. Please do not rely on receiving responses outside of weekdays between 9am & 5pm.

1.2 Content

- HW includes exercises that apply course concepts to novel settings, and open-ended questions that challenge you to **synthesize** and **build upon** course concepts. This is just like a language class! You learn the grammar, vocabulary, and structure needed to express your own ideas (as opposed to memorizing every sentence you might ever want to say).
- "Optional" exercises won't be graded, but you're strongly encouraged to try them!

1.3 Timing & flexibility

- You're strongly encouraged to hand in HW by 5pm, but it's technically due by 11:59pm.
- Extensions
  - Each student can have a 3-day extension on up to **3** of 7 HW, including weekends. Since this HW is due on a Thursday, a 3-day extension would make it due on or before Sunday.
  - Except in rare extenuating circumstances, HW that exceeds these extension opportunities will not be graded or accepted for credit. This policy is in place to provide some flexibility while ensuring that you stay on track to succeed in this course.
  - You don't have to reach out if you use one of your extensions. Moodle tracks this. I will also record your use of extensions in the "Homework extension" assignment.
- Your 1 lowest HW score will be dropped at the end of the semester.

1.4 Academic integrity

Review and stick to the academic integrity expectations in the syllabus. For *example*, you may *not*:

- use any materials from past iterations of STAT 253
- use any online solutions manuals or forums where you post HW related questions
- pass off another person's work as your own. (You're encouraged to discuss HW with classmates but all submitted work must be your own, from the words to the code.)

1.5 Grading

1.5.1 Passing homework

You will make mistakes and your HW won't be perfect. And that's ok! Instead of every mistake chipping away at your grade, HW will be graded pass / fail. To pass a HW, you must meet the following goals:

1. Your HW is handed in on time, or within your extension window.
2. Your work is reproducible and presented professionally. Specifically:
   - Use the provided Rmd template.
     - Update the author (your name).
     - You can make necessary modifications to this template (eg: add answers, R chunks), but cannot make any deletions or changes to the structure.
   - Include all RStudio code and output that's relevant to the HW exercises.
     Omit any RStudio code that's *not* relevant to the HW exercises.
   - Submit your **knit html** (not Rmd) file to the correct HW link on Moodle.
3. Your answer to *each* exercise is correct or mostly correct. The following are required to earn a "correct" score:
   - Answer is correct and complete.
   - Answer is supported with appropriate evidence (e.g. R code and output).
   - Discussions are based in the context of the exercise, not general definitions.
   - R code is well-formatted and organized.

   The following are required to earn a "mostly correct" score:
   - There are some mistakes, but you got more than ~75% correct.
   - You demonstrated an earnest attempt at each part of the exercise.

Your HW score will be recorded as follows on Moodle:

points	passing status	detail
3	passed	All exercises were correct.
2	passed	There were some notable mistakes, but all exercises were either mostly correct or correct. A 2 or 3 are not different in terms of your grade. A 2 merely communicates that you should revisit your HW.
1	did not YET pass	At least 1 exercise had significant errors or gaps. Yet HW meets the below criteria, thus can be revised and resubmitted.
0	did not pass	HW was not submitted on time / within the extension window. OR HW was submitted with significant gaps / errors, and does not meet the below criteria for resubmission.

1.5.2 Revisions

Revision is critical to learning! When mistakes are made (which they will be), revision provides an opportunity to reflect, improve, & deepen your understanding. You have the opportunity to revise and resubmit this HW up to one time **IF AND ONLY IF**:

- You handed the HW in on time or within your extension window; and
- Your work demonstrates a reasonable attempt to answer each question.

Details:

- Revisions must be handed in within 1 week of receiving feedback on your original HW submission. You must submit these revisions on Moodle.
- There are no extensions for revisions. It's important to your learning and progress in the course to do this as soon as possible.

1.5.3 Reflection

Solutions will be posted after the HW extension window has elapsed. Be sure to review these solutions, even if you passed the HW.

2 Exercises

Homework 3 goals

- Implement our new model building technique, LASSO, and compare it to least squares.
- Keep practicing foundational skills such as data wrangling and model evaluation.

Homework 3 context

What makes a song popular? Can we predict popularity using only acoustic properties (eg: how fast and loud a song is)? To this end, we'll analyze 247 songs that play on Spotify. Along with the `popularity` of each song, the data contains lots of acoustic variables. To learn more about the acoustic variables, and check out the orginal source on Tidy Tuesday, click here.

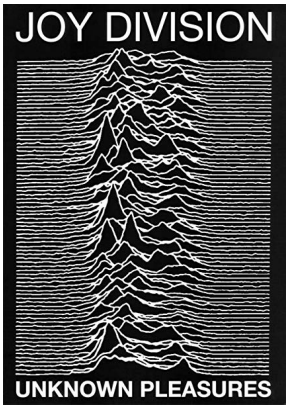
```
# Load packages
# (You might need to install ggridges)
library(ggridges) # for joy plots
library(tidymodels)
library(tidyverse)

# Load data
music <- read.csv("https://ajohns24.github.io/data/spotify_new.csv")
```

(1) **Prepare & acquaint yourself with the data**

a. Calculate the range of the observed `popularity` outcomes, from the minimum to the maximum.

b. Print out only the artist, song title, and popularity score for the 6 most popular songs in the data set.

c. *Joy Division* is perhaps the *only* band to have a data visualization tool named after them. The "Joy plot" is inspired by the band's album cover:



Just for fun, construct a joy plot for the popularity of the artists in our sample. Comment on two artists you find interesting (either because of their data or because you know them).

```
ggplot(music, aes(x = popularity, y = artist)) +
  geom_density_ridges() +
  theme_ridges()
```

d. Create a new data set `music_sub` which removes the following features which we either can't or won't use as predictors of popularity: `title` (can't use), `album_name` (won't use), `album_release_date` (won't use), `artist` (won't use).[1]

(2) **LASSO (Part 1)**

A small record label exec asks us to build a predictive model of `popularity` using the available set of predictors in `music_sub`. They don't have a data team, hence prefer a simpler model. Let's start by trying a LASSO model. **IMPORTANT:** (1) Each time you run a random process, use `set.seed(253)`. (2) Support your answer to each part with R output.

a. Build a reasonable, *final* LASSO model. You should try a range of **100 possible values** between 10^{-5} and 10^{1} for the λ tuning parameter, and use 10-fold CV MAE to pick only **one** of these for your final model.
b. For reproducibility, report the exact value of λ used in your final LASSO model.
c. How many and which of the 12 original predictors remain in your final LASSO model?
d. Report *and* interpret the CV MAE for your final LASSO model. Explain whether this is "big" or "small", and support this with some context.

(3) **LASSO (Part 2)**

You tuned the LASSO algorithm above, picking an appropriate value of λ for your final LASSO model. Let's think more about this tuning process.

a. Construct a plot that illustrates the impact of λ on the 10-fold CV MAE (and possibly 10-fold CV R^2). IMPORTANT: Take personal note of where *your* λ falls into this plot, and convince yourself that we tried a reasonable range for λ.
b. Summarize the key themes of this plot in your own words.
c. Construct a plot that illustrates the impact of λ on the predictor coefficients. IMPORTANT: Take personal note of where *your* λ falls into this plot.
d. Summarize the key themes of this plot in your own words. IMPORTANT: You don't need to write this out but be sure to convince yourself that you understand what each number and line on this plot represents.

[1] Take Bayesian Stat or Correlated Data to learn about incorporating artist and album name!

(4) **Finalizing our analysis**

Results for the least squares alternative to the LASSO are included below:

```
ls_model %>%
  tidy()
## # A tibble: 17 x 5
##    term                estimate  std.error statistic     p.value
##    <chr>                  <dbl>      <dbl>     <dbl>       <dbl>
##  1 (Intercept)      78.2        14.5           5.40  0.000000165
##  2 genrelatin       12.9         5.34          2.41  0.0167
##  3 genrepop         13.4         4.16          3.22  0.00149
##  4 genrer&b         14.3         4.30          3.33  0.00103
##  5 genrerap          6.17        4.13          1.49  0.137
##  6 genrerock        13.8         4.84          2.85  0.00472
##  7 danceability     -0.0717      0.0889       -0.807 0.421
##  8 energy           -0.227       0.113        -2.01  0.0454
##  9 loudness          1.48        0.580         2.55  0.0116
## 10 mode             -2.02        2.33         -0.867 0.387
## 11 speechiness       0.166       0.125         1.33  0.186
## 12 acousticness      0.0464      0.0630        0.736 0.462
## 13 instrumentalness -0.0801      0.0603       -1.33  0.185
## 14 liveness          0.0581      0.0826        0.703 0.483
## 15 valence          -0.0550      0.0595       -0.923 0.357
## 16 tempo            -0.0147      0.0425       -0.347 0.729
## 17 duration_ms      -0.00000653  0.0000226    -0.288 0.773

ls_cv %>%
  collect_metrics()
## # A tibble: 1 x 6
##   .metric .estimator  mean     n std_err .config
##   <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 mae     standard    14.4    10   0.801 Preprocessor1_Model1
```

a. Keeping in mind the music exec's goals, which model would you pick: the least squares model or your final LASSO model? *Explain.* Support your explanation with specific context and results.

b. Is your chosen model *wrong*? (It's ok if it is – we won't fix it now.) Support your answer with appropriate evidence.

c. Based on the results of your chosen model, offer the music exec some advice on how to write the most popular song possible.
   - Use complete sentences. Think of the exec as your client.
   - Don't get into rigorous coefficient interpretations! This wouldn't be useful for the exec. Simply focus on what *increases* popularity and what *decreases* popularity.

(5) **OPTIONAL: Homework reflections**

You are strongly encouraged to reflect upon how Homework 3 went:

- What was challenging?
- With what topics or tools did you feel the most confident?
- How did you manage the homework: When did you start? Did you review the material before starting? Who did you work with? Did you ask questions in OH / Slack?
- How might you improve how you manage future homeworks?

(6) **OPTIONAL: Unit reflection**

We find ourselves at the end of the (short) unit on "Regression: Model Building"! Take some time to collect your notes and map out some ideas. Some things to think about:

- What was the main motivation / goal behind this unit?
- For each of the following algorithms, describe the steps, pros, cons, and comparisons to least squares:
  - best subset selection
  - backward stepwise selection
  - LASSO
- In your own words, define the following: parsimonious models, greedy algorithms, goldilocks problem.
- Review the new tidymodels syntax from this unit. Identify key themes and patterns.

(7) **OPTIONAL: Spotify + R**

You can directly play around with Spotify's API within R! If you're curious, follow the instructions here to import data into R related to any Spotify play list, artist, genre, etc.