MSB Final Project: The Genetics of Music

Kabir Gupta

May 21, 2021

Introduction

One of Tinbergen's four causes of behavior is development, or ontogeny: the question of how a behavior came to an organism in the first place. While this subject can delve deep into Mendellian inheritance and meiosis, or different forms of learning and different classifications of innate behaviors, the most basic and central question asked during the study of development is that of nature versus nurture: is the trait genetically inherited or environmentally acquired? Of course, traits are often not one or the other but will rather fall on a spectrum between the two; the purpose of this project is to study one such trait in hopes of determining whether it is influenced more by genes or environment. This trait is musical taste: what kinds of music do people like to listen to?

It is surprisingly common, in the researcher's own (anecdotal) experience, to find two friends who have vastly different music tastes, and the same applies to family members. This leads into the question of who makes more of an impact on your music tastes — friends will often attest to have virtually identical music taste, but the same could apply to family members; it only seems to vary from person to person and family to family. By gathering a large sample of data, however, it could be determined whether there is actually some sort of trend: an association between a person's music tastes and their friends'/family's.

It was expected that both family members and friends would have a significant impact on a person's music choices, although one might not necessarily outweigh the other: that is, there may not be a significant difference between the family's impact and the friends'. This is because people have very strong reasons to like similar music to both a family member and a friend, so that one relationship type should not be significantly closer than the other.

¹Special thanks to Lincoln Auster and Thomas Morford for code review, bug fixes and formatting advice.

Materials and Methods

A Google Forms survey was used to gather data for this study, and a Python script was used to analyze it (included in Appendix I). Each respondent was asked to select the musical genres that they enjoy out of 6 options: pop, jazz/blues, rock, country/folk, rap/hip hop, and classical. The respondent was then asked to decide whether they like or dislike each of 12 musical clips, out of which each genre was represented by two clips: Uptown Funk and Dead Girl in the Pool (pop), What a Wonderful World and The Thrill is Gone (jazz and blues), Bohemian Rhapsody and Hurt (rock), Old Town Road and I Walk the Line (country and folk), Rap God and Gangsta's Paradise (rap and hip hop), and finally Eine Kleine Nachtmusik and Duel of the Fates (classical).

For the purposes of this study, it was useful to be able to identify whether there were some genres that the respondent did not necessarily dislike, even if it was not something they thought they actively listen to. So, while the first part of the survey asked for what they thought they like, the second part of the survey was in an attempt to see what they might actually like. However, it's possible that somebody who generally likes a certain genre did not like the specific two clips that they had been asked about from that genre. So, the question about categories needed to be weighted more heavily than the questions about specific songs. Further, an overall system was needed for condensing the 2-part data that would be gathered — genre and music preference, both on nominal scales — into one number that could be used in statistical testing. Hence, the following scoring system was created, as seen in Figure 1.

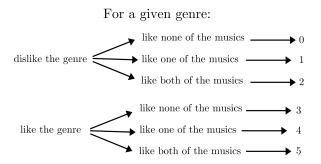


Figure 1: The procedure used to generate ordinal-scale scores for each respondent, for each genre.

This means that if a respondent says they enjoy a particular genre but doesn't like any of the musics, they can still walk away with a score of 3, whereas if they liked both musics but said they disliked the genre in general, they would only get a 2. These rankings are on

an ordinal scale, from 0 to 5 (where there are no defined intervals between ranks).

Each respondent was requested to provide the names of two friends and two family members. The family members were sent a slightly different survey, which was identical to the one for students except that it did not ask them for the names of their friends or family members. At the end of data collection, each respondent was matched up to their two family members and two friends to perform statistical analysis. If not all of the family members and friends listed had responded to the survey, only the ones whose responses were present were tested against the student.

Data

There were two main variables being compared in this study: relationship type (family member or friend), on a nominal (binary) scale; and genre rankings that demonstrate the degree to which someone enjoys a particular genre, on an ordinal scale.

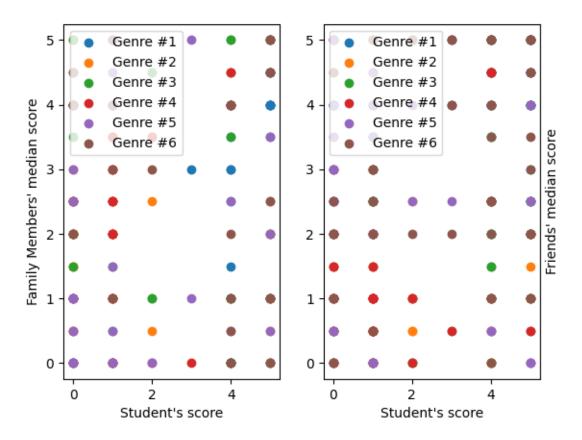


Figure 2: Scatter plots displaying the data gathered. At left, each student's score is plotted against the median of their family members' scores for each genre; at right, each student's score is plotted against the median of their friends' scores for each genre.

The scatter plots above display the results of data collection, spread across genres. The raw data gathered during this study is included in Appendix II.

A Wilcoxon test was used for the primary statistical analysis. Out of 49 tests conducted

for a difference between the students' rankings and their family members', 9 were statistically significant at a 90% confidence level (a=0.10), and 5 were significant at a 95% confidence level (a=0.05). For the friends, 14 tests came out significant at the 90% confidence level, and 7 at the 95% confidence level, out of a total of 79 tests. Overall, 18.4% of tests conducted for family members came back significant, compared to 17.7% of tests for friends.

The follow-up chi-square test for association had an insignificant result, meaning that the null hypothesis must be accepted: there is no association between relationship type and similarity of rankings. The test statistic χ^2 was calculated to be 3.389771795679717 with 9 degrees of freedom; p=0.9468207928306335.

An SRCC was also calculated for each relationship type for each genre. For family members, the correlation coefficients were $r=0.48,\,0.05,\,-0.10,\,0.13,\,0.21,\,$ and -0.15 respectively. For friends, they were $r=0.21,\,0.21,\,0.07,\,0.04,\,-0.03,\,$ and 0.06 respectively. Three of the coefficients overall (two on the family side and one on the friends side) were negative; all the rest were positive. Almost all of the correlation coefficients were quite weak (below 0.1); however, the correlation coefficient for Genre 1 (Pop) for family members could be classified as of moderate strength. Apart from the aforementioned coefficient, which had a p-value p=0.00679 (highly statistically significant), all the other coefficients were insignificant.

Finally, a t-test for two correlated samples was conducted between the family members' and friends' SRCCs to test for a significant difference in the correlation coefficients. The test statistic reported was t=0.14. The critical value for the t-distribution at a 95% confidence level with 5 degrees of freedom is 2.57, and the test statistic found does not exceed this. The p-value was only p=0.90, so the result was insignificant. Thus, we fail to reject the null hypothesis that family members and friends both influence people's musical choices to approximately the same extent.

Statistical Analysis

The most preliminary technique used in analyzing the collected data was assigning each individual an ordinal-scale ranking for each genre (in order to have two easily comparable populations). The specifications of how this was done have been provided in the Methods section. Once each student, family member and friend was assigned a set of 6 ranks (as there were 6 genres involved), a comparison of medians was conducted between the data sets. Because the data was on an ordinal scale, the statistical test had to be nonparametric. Further, the data was paired because each student was related to their corresponding family member or friend in some way. Consequently, a Wilcoxon test was used to test for a significant difference between each student's rankings and their family members', and between each student's rankings and their friends'. The results of these tests are summarized in the previous section.

The p-values generated by the Wilcoxon tests were sorted into an aggregated frequency table (class borders 0-0.1, 0.1-0.2, 0.2-0.3, and so on up to 1.0). This was done to each set of tests (family and friends), allowing for the construction of the contingency tables below (Tables 1 and 2). A chi-square test for association was conducted to find if there was an association between to what extent two people differ in music choice, and how the two people are related (family or friend).

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Family	9	11	3	5	6	1	3	6	2	3
Friend	14	16	8	4	10	5	4	9	5	4

Table 1: Contingency table of observed values for χ^2

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Family	8.80	10.34	4.21	3.45	6.13	2.30	2.68	5.74	2.68	2.6
Friend	14.20	16.66	6.79	5.55	9.88	3.70	4.32	9.26	4.32	4.32

Table 2: Contingency table of expected values for χ^2

Although the chi-square test returned an insignificant result, it was still worth calcualting a Spearman Rank Correlation Coefficient to get a sense of how strong the correlation between each pair of two people was for each genre. This gave a set of twelve SRCCs, one for each genre and for each relationship type (friend/family). The SRCC comparison tables were of the format as in Table 3 (only a part of the first one is included here, as they were all

similarly structured.)

x: student's ranking	y: family member's ranking
0	4
1	0
1	0

Table 3: SRCC table of values for students vs family members for Genre 1, Pop (truncated after row 3).

The twelve correlation coefficients that were found (six for each relationship type) have been reported in the previous section. One last statistical test was conducted, the t-test for two correlated (paired) samples. A significant result here would have meant that one relationship type (be that family member or friend) has significantly higher correlations between two people than the other. The two samples consisted of the set of all SRCCs for family members, and the set of all SRCCs for friends (thus n=6). So, each sample was matched because every SRCC for the family members applied to the same genre as the corresponding SRCC for the friends. Further, a parametric test could be used because correlation coefficients lie on a ratio scale (with an absolute zero) and there was no reason to suspect major skew in the data. The results of the two-sample t-test were reported in the previous section as well.

Discussion

Conducting the Wilcoxon tests for each student revealed that there were some cases where there was a significant difference between family members, and likewise for friends. However, when all of those cases were tallied up in a chi-square test, there did not appear to be a significant association between relationship type and music choices. While calculating the SRCCs supported this conclusion for the most part, one of the coefficients did suggest that family members may exert a special influence on students' choices when it comes to pop music based on the correlation coefficient alone. However, when a t-test was run between the SRCCs to check for a significant difference, the result was clear: neither relationship has greater power over a person's musical choices than the other.

The data gathered thus supports the original hypothesis that an environmental influence would not outweigh a genetical influence on a person's music choices, and vice versa. However, there are two ways these results could have been made more reliable:

- A larger sample size always helps increase the power of statistical analysis, and can thus catch smaller differences in a population that a small sample size cannot. The sample size for this study was 76 students and 65 family members; while that's not a tiny sample size, this study would definitely have been more powerful if (a) more people participated, but even more so (b) for every student who participated, both family members and both friends were able to fill out the respective surveys.
- The twelve songs selected to "represent" each genre may not have been very representative of each genre. For instance, "Hurt" is a heavy metal song that even an avid rock listener might not find as appealing as a pure rock song. "Eine Kleine Nachtmusik," while popular, is overused to the point of many classical enthusiasts not enjoying it as much as some Brahms or Shostakovich; and "Duel of the Fates" is a vocalized theme from the *Star Wars* soundtrack, so that it can technically be categorized as classical but when compared to Bach or Beethoven seems vastly different. These oversights were due to the researcher's own relative lack of musical knowledge (which has slightly improved over the course of this study).

A redo of this study could have the potential to be more successful if more time was spent to take as large a sample size as possible — for instance, if everybody in one school filled it out, it might only provide 500 people, but at least each person's friend would have (hopefully) filled it out too until it cycled back so that every person can be tested against four other people — and to use music clips that were much more representative of the genre (and likely to be liked by somebody who listens to the genre often).

Another possible flaw in this research was that it assumes that family members are the "genetic" factor, and friends are the "environmental". While it's true that there's usually no genetic relationship between two friends, the problem is that one's family is not only who they share the greatest amount of their DNA with, but also who they have usually grown up around and been brought up with. This means that a family member might encapsulate both the genetic and the environmental factor. Thus, this study cannot really conclude whether music has a large genetic part, because it is quite possible that the overlaps found between students and their family members were due to a shared environment — in some cases, people may even share much more of their environment with family members than with friends. To rectify this, a completely different study might need to be carried out, maybe one connecting genetically related people who have not been in the same environment (cousins? siblings who were separated at birth? long-lost parents or children? clones?) might be more powerful in revealing whether there is more of a genetic and an environmental factor at play.

References

A Google Forms survey was used in this study, and the data was linked into a Google Sheet spreadsheet. The following twelve songs were used during the study; the relevant portions were clipped out (10 seconds) for survey-takers to listen to.

- 1. "Uptown Funk", Mark Ronson, 00:58 to 1:08. https://youtu.be/OPf0YbXqDm0
- 2. "Dead Girl in the Pool", girl in red, 00:34 to 00:44. https://youtu.be/Pzq4TEU-wHo
- 3. "What a Wonderful World", Louis Armstrong, 00:17 to 00:27. https://youtu.be/ VqhCQZaH4Vs
- 4. "The Thrill is Gone", B.B. King, 00:34 to 00:44. https://youtu.be/oica5jG7FpU
- 5. "Bohemian Rhapsody", Queen, 00:02 to 00:12. https://youtu.be/fJ9rUzIMcZQ
- 6. "Hurt", Nine Inch Nails, 4:01 to 4:11. https://youtu.be/OvoTktdpIiI
- 7. "Old Town Road", Lil Nas X, 00:39 to 00:49. https://youtu.be/r7qovpFAGrQ
- 8. "I Walk the Line", Johnny Cash, 00:25 to 00:35. https://youtu.be/J5126CibNsk
- 9. "Rap God", Eminem, 00:25 to 00:35. https://youtu.be/XbGs_qK2PQA
- 10. "Gangsta's Paradise", Coolio, 00:57 to 1:07. https://youtu.be/fP076Jlnz6c
- 11. "Eine Kleine Nachtmusik", Wolfgang Amadeus Mozart, 00:05 to 00:15. https://youtu.be/oy2zDJPIgwc
- 12. "Duel of the Fates", John Williams, 1:03 to 11:13. https://youtu.be/C2XUJ5PWg-8

For the Python script, the SciPy Statistics library (scipy==1.6.3) was used to perform statistical analyses. The Pandas library (pandas==1.2.4) was used to import data from a CSV (exported from Google Sheets) into the Python script. Finally, the Matplotlib library (matplotlib==3.4.2) was used to draw scatter plots from the data collected.

Appendix I

The following scripts were used to analyze the data gathered. Main program:

```
#!/usr/bin/env python3
  def findmatches(student, studs, fams):
      # assign family members and friends for a given student
      fam0 = "" # an empty string by default
      fam1 = ""
      friend0 = ""
      friend1 = ""
      for friend in studs:
          if student.friends[0] == friend.name:
              friend0 = friend
          if student.friends[1] == friend.name:
              friend1 = friend
      for family in fams:
          if student.fam[0] == family.name:
              fam0 = family
17
          if student.fam[1] == family.name:
18
              fam1 = family
19
      return (fam0, fam1, friend0, friend1)
21
22
  def wc(studs, fams):
      # initialize arrays to hold the results of the WC tests
24
      fam_results = []
25
      friend_results = []
26
      for index in range(len(studs)):
28
          student = studs[index]
          fam0, fam1, friend0, friend1 = findmatches(student, studs, fams)
30
          if fam0 != "": # fam0 exists
32
              # run tests between student and family member 0
              if student.ranks == fam0.ranks:
34
                   print(
                       "\{\} \{\} and \{\} \{\} (family member) have the same
     rankings. p=0".format(
                           student.name[1], student.name[0], fam0.name[1],
     fam0.name[0]
```

```
p = 0
40
               else:
                   w, p = wilcoxon(student.ranks, fam0.ranks)
42
                   print(
43
                        "The Wilcoxon test statistic between {} {} and {} {}
44
     family member) is W={} (p={}).".format(
                            student.name[1],
45
                            student.name[0],
46
                            fam0.name[1],
47
                            fam0.name[0],
48
49
                            w,
                            p,
50
                        )
               fam_results.append(p)
53
54
           if fam1 != "": # fam1 exists
               # run tests between student and family member 1
56
               if student.ranks == fam1.ranks:
57
                   print(
58
                        "\{\} {} and {} {} (family member) have the same
     rankings. p=0".format(
                            student.name[1], student.name[0], fam1.name[1],
60
     fam1.name[0]
                        )
61
                   )
62
                   p = 0
63
               else:
                   w, p = wilcoxon(student.ranks, fam1.ranks)
65
                   print(
66
                        "The Wilcoxon test statistic between {} {} and {} {}
67
     family member) is W={} (p={}).".format(
                            student.name[1],
68
                            student.name[0],
                            fam1.name[1],
70
                            fam1.name[0],
                            w,
73
                            р,
                        )
74
                   )
75
               fam_results.append(p)
76
77
           if friend0 != "": # friend0 exists
               # run tests between student and friend 0
79
               if student.ranks == friend0.ranks:
80
                   print(
81
                        "\{\} {} and {} {} (friend) have the same rankings. p=0"
82
      .format(
                            student.name[1],
83
                            student.name[0],
84
                            friendO.name[1],
                            friendO.name[0],
86
```

```
р
                      = 0
89
                else:
90
                    w, p = wilcoxon(student.ranks, friend0.ranks)
91
                    print(
92
                         "The Wilcoxon test statistic between {} {} and {} {}
93
      friend) is W={} (p={}).".format(
                             student.name[1],
94
                             student.name[0],
95
                             friendO.name[1],
96
                             friendO.name[0],
97
98
                             w,
                             p,
99
                         )
100
101
                friend_results.append(p)
           if friend1 != "": # friend1 exists
104
                # run tests between student and friend 1
                if student.ranks == friend1.ranks:
106
                    print(
                         "\{\} {} and \{\} {} (friend) have the same rankings. p=0"
108
      .format(
                             student.name[1],
109
                             student.name[0],
                             friend1.name[1],
                             friend1.name[0],
                         )
113
                    )
                    p = 0
115
                else:
116
                    w, p = wilcoxon(student.ranks, friend1.ranks)
117
118
                    print(
                         "The Wilcoxon test statistic between {} {} and {} {} (
119
      friend) is W={} (p={}).".format(
                             student.name[1],
120
                             student.name[0],
121
                             friend1.name[1],
                             friend1.name[0],
123
                             w,
124
125
                             p,
                         )
126
                    )
127
128
                friend_results.append(p)
130
       return (fam_results, friend_results)
131
132
133
  def chi2(group1, group2):
134
       # conduct chi-square test for association
135
       g1 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
       g2 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
137
     for x in range(10):
```

```
for p in group1:
                if p \ge 0.1 * x and p < 0.1 * (x + 1):
140
                    g1[x] += 1
141
           for p in group2:
142
                if p > 0.1 * x and p < 0.1 * (x + 1):
143
                    g2[x] += 1
144
       for p in group1:
145
           if p == 1:
146
                g1[9] += 1
147
       for p in group2:
148
           if p == 1:
149
                g2[9] += 1
151
       chi2observed = [g1, g2]
       print(
153
           "Here are the observed frequencies of each type of result for
      family members and friends, respectively:"
       print(chi2observed)
156
       X2, p, df, ex = chi2_contingency(chi2observed)
157
       print(
158
159
           "Here are the expected frequencies of each type of result for
      family members and friends, respectively:"
160
       print(ex)
161
       print(
162
           "The result of the Chi-Square Test for Association is test
163
      statistic X^2 = \{\} with \{\} degrees of freedom, which gives us a p-value
       of {}.".format(
                X2, df, p
164
165
166
       return p
167
168
  def srcc(students, families):
170
       famccs = []
171
       friendccs = []
       fig, (famplot, friendplot) = plt.subplots(1, 2)
173
       for genre in range(6):
174
           studentranksa = []
                                # for fam
           studentranksb = []
                                # for friend
176
           famranks = []
177
           friendranks = []
179
           for student in students:
180
                fam0, fam1, friend0, friend1 = findmatches(student, students,
181
      families)
182
                studentranksa.append(student.ranks[genre])
183
                if fam0 != "" and fam1 != "": # both of them were found
184
                    famranks.append(
                         (fam0.ranks[genre] + fam1.ranks[genre]) / 2
186
                    ) # median not mean
187
```

```
elif fam0 != "": # only fam0 was found
                    famranks.append(fam0.ranks[genre]) # pick the only one we
189
       have
               elif fam1 != "": # only fam1 was found
190
                    famranks.append(fam1.ranks[genre])
191
               else: # no family members were found
192
                    studentranksa.remove(
193
                        student.ranks[genre]
194
                       # we can't use this student's data
196
               studentranksb.append(student.ranks[genre])
197
               if friend0 != "" and friend1 != "": # repeat same for the
198
      friends
                    friendranks.append(
199
                        (friend0.ranks[genre] + friend1.ranks[genre]) / 2
200
                     # median again
201
               elif friend0 != "":
202
                    friendranks.append(friend0.ranks[genre])
               elif friend1 != "":
204
                    friendranks.append(friend1.ranks[genre])
205
               else:
206
207
                    studentranksb.remove(student.ranks[genre])
208
           print("The student ranks (x1) for Genre #%i were: " % (genre + 1),
209
       end="")
           print(studentranksa)
210
           print("The family ranks (y1) for Genre #%i were: " % (genre + 1),
211
      end="")
           print(famranks)
212
           print("The student ranks (x2) for Genre #%i were: " % (genre + 1),
213
       end="")
           print(studentranksb)
214
           print("The friend ranks (y2) for Genre #%i were: " % (genre + 1),
215
      end="")
           print(friendranks)
216
217
           gen_name = "Genre #%i" % (genre + 1)
218
219
           famplot.scatter(
220
               studentranksa, famranks, label=gen_name
221
              # add stud and fam to scatter plot
222
           friendplot.scatter(
223
               studentranksb, friendranks, label=gen_name
224
              # add stud and friend to scatter plot
225
226
           r, p = spearmanr(studentranksa, famranks)
227
           print(
228
               "The correlation coefficient between students and family
229
      members for Genre #%i was %2.5f (p=%2.5f)"
               % ((genre + 1), r, p)
230
           )
231
           famccs.append(r)
           r, p = spearmanr(studentranksb, friendranks)
233
234
           print(
```

```
"The correlation coefficient between students and friends for
      Genre #%i was %2.5f (p=%2.5f)"
               % ((genre + 1), r, p)
236
           )
237
           friendccs.append(r)
238
230
       famplot.legend(loc="upper left") # create legends
240
       famplot.set_xlabel("Student's score")
241
       famplot.set_ylabel("Family Members' median score")
242
       friendplot.legend(loc="upper left")
243
       friendplot.set_xlabel("Student's score")
244
       friendplot.set_ylabel("Friends' median score")
245
       friendplot.yaxis.set_label_position("right") # so that it has room to
246
       be seen
       plt.savefig("scatter_plots.png") # export to a png
247
248
       return famccs, friendccs
249
251
  def main():
252
       # create lists to hold student and family objects:
253
254
       studs = []
       fams = []
255
       print(students)
256
       print(families)
257
       print(
258
           "CSVs were successfully imported into the pandas dataframe;
259
      proceeding to creating objects for each individual."
260
261
       # add a student/family object for each row in the dataframe
262
       for row in range(students.shape[0]):
263
           studs.append(Student.at_row(row))
                                                 # create students
       for row in range(families.shape[0]):
265
           fams.append(Family.at_row(row)) # create family
266
267
       print(
268
           "Student/Family objects were successfully created; running
269
      Wilcoxon tests between each individual and their friends/family members
      . 11
       )
270
271
       # run wilcoxon tests
272
       genetic, environmental = wc(studs, fams)
273
       print()
274
       print("Here are the results of the Wilcoxon Tests for the family
275
      members:")
       print(genetic)
276
       print("Here are the results of the Wilcoxon Tests for the friends:")
277
       print(environmental)
278
       print("Here is the most common result of the tests for the family
279
      members:")
       print(multimode(genetic))
280
       print("Here is the most common result of the tests for the friends:")
```

```
print(multimode(environmental))
283
       p = chi2(genetic, environmental)
284
       if p > 0.5:
285
           print(
286
               "WARNING: There is not a statistically significant association
287
       between relationship type and similarity of music choice. Following
      results may be very unreliable."
           )
288
       print()
289
       genecc, envcc = srcc(studs, fams)
290
291
       print("Genetic correlation coefficients, as a list: " + str(genecc))
292
       print("Environmental correlation coefficients, as a list: " + str(
293
      envcc))
294
       print(
295
           "The average correlation coefficient for a genetic relationship is
           + str(mean(genecc))
297
       )
       print(
           "The average correlation coefficient for an environmental
300
      relationship is "
           + str(mean(envcc))
301
302
303
304
       t, p = ttest_rel(genecc, envcc) # paired t-test for two samples
305
       print(
306
           "The paired t-test for a difference between genetic and
307
      environmental correlation coefficients gave the test statistic t="
           + str(t)
308
           + " "
309
       print("The p-value was " + str(p) + ".")
311
313
  if __name__ == "__main__":
314
       from scipy.stats import (
315
           wilcoxon,
316
           chi2_contingency,
317
           spearmanr,
318
           ttest_rel,
319
       ) # needed for statistical testing
320
       from statistics import mean, multimode # needed for descriptive
321
      statistics
       import matplotlib.pyplot as plt # needed for graphing SRCC data
322
       from people import * # needed dataframes & classes from ./people.py
323
324
      main()
325
```

Wrapper script (using POSIX sh):

```
#!/usr/bin/env sh
```

```
3 for csv in private/*.csv; do
          # delete timestamp column if present
          if [ "$(head -n1 < "$csv" | cut -d ',' -f 1)" = "Timestamp" ];
     then
                  cp "$csv" "${csv}.old" #make backup
6
                  cut -d ',' -f 2- <"${csv}.old" >"$csv"
                  rm "${csv}.old" #rm backup
                  printf 'Removed timestamps from %s\n' "$csv"
9
          fi
          # serialize names of genres to integer values
          sed -i '' 's/Pop/1/g' "$csv" && printf "replaced 'pop' with '1' in
13
      %s\n" "$csv"
          sed -i '' 's/Jazz\/Blues/2/g' "$csv" && printf "replaced 'jazz/
14
     blues' with '2' in %s\n" "$csv"
          sed -i '' 's/Rock/3/g' "$csv" && printf "replaced 'rock' with '3'
     in %s\n" "$csv"
          sed -i '' 's/Country\/Folk/4/g' "$csv" && printf "replaced '
16
     country/folk' with '4' in %s\n" "$csv"
          sed -i '' 's/Rap\/Hip Hop/5/g' "$csv" && printf "replaced 'rap/hip
      hop' with '5' in %s\n" "$csv"
          sed -i '' 's/Classical/6/g' "$csv" && printf "replaced 'classical'
18
      with '6' in %s\n" "$csv"
          sed -i '' 's/Dislike/0/g' "$csv" && printf "replaced 'dislike'
19
     with '0' in %s\n" "$csv"
          sed -i '' 's/Like/1/g' "$csv" && printf "replaced 'like' with '1'
     in %s\n" "$csv"
          printf 'finished working on %s\n\n' "$csv"
22 done
23 python3 main.py
25 printf '\nfinished running script, exiting now\n'
```

Class definitions for students and families:

```
#!/usr/bin/env python3
2
3 from pandas import read_csv # necessary for importing csv data as a
     pandas dataframe
5 students = read_csv("private/Students.csv") # initialize dataframe for
     students
6 families = read_csv("private/Family.csv") # and one for families
9 class Student:
      def __str__(self):
10
          return (
              str(self.name)
12
              + "\n"
              + str(self.ranks)
14
              + "\n"
              + str(self.fam)
16
              + "\n"
```

```
+ str(self.friends)
          )
19
20
      @classmethod
2.1
      def at_row(cls, n: int):
22
          row = [student for student in students.iloc[n]]
23
24
          return cls(row)
25
      # Construct a Student from a row of data. This row is expected to be
26
      # out in (first name, last name, ... musics ..., family member a
27
     first name,
      # family member a last name, family number b first name, family number
2.8
      # last name, friend a first name, friend a last name, friend b first
29
     name,
      # friend b last name).
30
      def __init__(self, row):
          self.name = (
32
              row[0],
33
               row[1],
34
          )
36
          self.fam = [
37
               (row[15], row[16]),
               (row[17], row[18]),
39
40
41
          self.friends = [
               (row[19], row[20]),
43
               (row[21], row[22]),
44
          ]
45
          self.ranks = [-1 for n in range(1, 7)]
47
          # generate & store rankings
49
          genres = row[2] # temporarily store genre selections
          musics = [row[n] for n in range(3, 15)]
          # Each genre is represented by a number (1=pop, 2=jazz & blues,
53
     etc).
          for genre in range(1, 7):
54
               if str(genre) in genres: # liked the genre
                   self.ranks[genre - 1] = (
                       3 + musics[genre * 2 - 2] + musics[genre * 2 - 1]
57
58
               else: # disliked the genre
59
                   self.ranks[genre - 1] = musics[genre * 2 - 2] + musics[
60
     genre * 2 - 1]
61
  class Family: # I'll have one object of this type for each participating
     family member; identical to Student but lacks fam[] and friends[]
  def __str__(self):
```

```
return str(self.name) + "\n" + str(self.ranks)
66
      @classmethod
      def at_row(cls, n: int):
68
          row = [family for family in families.iloc[n]]
69
          return cls(row)
70
      # Construct a Family from a row of data. This row is expected to be
72
     laid
      # out in (first name, last name, genres, musics ...).
73
      def __init__(self, row):
74
          self.name = (
              row[0],
76
              row[1],
          )
78
          self.ranks = [-1 for n in range(1, 7)]
80
          # generate & store rankings
82
          genres = row[2] # temporarily store genre selections
83
          musics = [row[n] for n in range(3, 15)]
84
          # Each genre is represented by a number (1=pop, 2=jazz & blues,
86
     etc).
          for genre in range(1, 7):
87
               if str(genre) in genres: # liked the genre
88
                   self.ranks[genre - 1] = (
89
                       3 + musics[genre * 2 - 2] + musics[genre * 2 - 1]
90
              else: # disliked the genre
92
                   self.ranks[genre - 1] = musics[genre * 2 - 2] + musics[
93
     genre * 2 - 1]
```

Appendix II

On the following two pages are presented the raw data collected over the course of study. In order to protect the privacy of those participating in the study, respondents' first and last names have been entirely ommitted, and their friends' and family members' names have been replaced with the unique timestamps associated with that person's response. The first table displays raw data for students (including friends), and the second for family members.

3/30/3021 18-25-03 Pop Bock Classical	Like	Distiles	Like	Distilier	Like	Dishler	Dishler	Like	Dishley	Diubles	Like	Like	3/31/3021 20:35:02	3/31/2021 20:35-02	3/31/2021 20:34-23	3/31/3021 30:34/23	5/17/2021 12:54:35	5/17/2021 12:54:35	3/30/2021 19/08/20	3/30/3021 19:08:30
3/30/2021 19:04:10 Pop. Rock, Classical	Dislike	Libr	Like	Distike	Like		Like	Distiler	Dishilar	Like		Didike	5/9/2021 22:40:12	5/9/2021 22:40:12	10, 10, 10, 10, 10, 10, 10, 10, 10, 10,	m, m, miss stratus		5/9/2021 12:54:35	5/9/2021 22:45:20	5/9/2021 22:45:20
3/30/2021 19:08:20 Pop. Jazz/Blues Ran/Hin Hon. Classical	Like		Like	Disting	Like		Like	Distile	Like			Like	3/30/2021 22:40:12		3/30/2021 19:11:45	3/30/2021 19:11:45	5/17/2021 12:54:35			3/9/2021 22:43:20
3/30/2021 19:08:20 Pop. Jazz/Elizes, Rap/Hip Hop, Classical 3/30/2021 19:16:50 Pop. Rap/Hip Hop. Classical	Like	Like		Distike	Like	Distilor	Like	Distile	Lifee	Life			3/30/2021 19:48:57	3/30/2021 19:48:57	3/30/2021 19:11:45	3/30/2021 19:11:45			3/27/2021 13:39:39	3/27/2021 13:30:39
	Like	Like	Like	Dulke	Disting	Lifer	Distile	Distilie	Lifee	Like	Distike	Distike					4/11/2021 18:49:19	4/11/2021 18:49:19		
3/30/2021 19:38:17 Pop. Rap/Hip Hop. Classical	Like	Like	Dieblor	Dislike	Distilor	Dishlor	Like	Distike	Like	Like	Like	Dislike	5/10/2021 8:33:11	5/10/2021 8:33:11						
3/30/2021 19:43:52 Pop. Classical	Dislike	Dishler	Dielike	Dislike	Dishlor	Dishke	Dislike	Distilier	Didde	Diebler	Dislike	Distrike					3/30/2021 19:47:44	3/30/2021 19:47:44	3/30/2021 20:49:59	3/30/2021 20:49:59
3/30/2021 19:47:44 Pop. Country/Folk, Rap/Hip Hop. Classical	Like	Like	Lifer	Like	Like	Lifer	Dodike	Dislike	Life	Like		Distilie	5/13/2021 10:04:34	5/13/2021 10:04:34	3/30/2021 20:01:45	3/30/2021 20:01:45	3/30/2021 19:43:52		3/30/2021 20:49:59	3/30/2021 20:49:59
3/30/2021 19:57:52 Pop. Rap/Hip Hop	Distiller	Distilion	Dielike	Dislike	Like	Dishlor	Like	Distrike	Like	Like	Dislike	Dislike	5/10/2021 7:58:09	5/10/2021 7:58:09	5/10/2021 8:32:01	5/10/2021 8:32:01	5/17/2021 12:54:35	5/17/2021 12:54:35	5/9/2021 22:22:38	5/9/2021 22:22:38
3/30/2021 20:30:10 Pop. Rock, Classical	Like	Like	Lifer	Dislike	Like	Dishke	Dislike	Distilie	Like	Dichlo	Like	Distilie	5/10/2021 6:44:47	5/10/2021 6:44:47	3/30/2021 20:44:13	3/30/2021 20:44:13				
3/30/2021 20:49:59 Rock	Dislike	Dislike	Lifer	Dislike	Like	Lifer	Dislike	Like	Distiler	Dielike		Distike	3/30/2021 21:56:23	3/30/2021 21:56:23			3/30/2021 19:43:52	3/30/2021 19:43:52	3/30/2021 19:47:44	3/30/2021 19:47:44
3/31/2021 1:00:09 Pop. Jazz/Blues, Rock, Country/Folk, Rap/Hip Hop. Classical	Like	Like	Like	Like	Distiler	Didlo	Like	Like	Like	Like	Like	Like					5/12/2021 22:07:21	5/12/2021 22:07:21	5/10/2021 0:22:24	5/10/2021 0:22:24
3/31/3091-047-05 Rock Closeles	Like	District	Lifer	Distrike	Like	Dishler	Divities	Distribut	Dishler	Dishler	Like	1 three			5/10/2021 16:28:12	5/10/2021 16/28 12	3/30/2021 19:08:20	3/30/2021 19:08:20	4/20/2021 19:44:09	4/20/2021 19:44:09
3/31/2021 7:56:24 Pop. Jazz/Blues Country/Folk Classical	Like	Distilia	Like	Distilier	Like	Dishler	Like	Dishike	Dishler	Dishler	Like	Distribu			37.737.2027.7032032	37.107.000.1000.00	3, 34, 434, 13, 31, 44	3,34,404,153344	3/31/3021 23 14:02	3/31/2021 23:14:02
3/31/2021 11:29:15 Pop. Rock, Ran/Hin Hop	Like		Life	Distilier	1.0	Like	Distiles	Distilie	Distilier	Like		Diddle	3/31/2021 11:36:00	3/31/2021 11:36:00	3/31/2021 13:59:28	3/31/2021 13:59:28	3/31/2021 15:45:07	3/31/2021 15:45:07	3/31/2021 23:14:02	3/31/2021 23:14:02
3/31/2021 11:27:15	Like	Distinct	Like	Distille	1.160	Dishlor	Distan		Distant	Diables		Like	3/31/2021 11:36:00	3/31/2021 11:39399	3/31/2021 13:00:28	3/31/2021 13:59:28	3/31/2021 15:45:00		3/31/2021 2/31492	3/31/2021 6:47:05
		Dielike			Like		Distike	Like	Distino	Dichic	Like									
3/31/2021 12:45:53 Pop 3/31/2021 12:45:53 Pop 3/31/2021 12:45:53 Pop	Like	Disting	Like	Like	Like	Dishlor	Like	Distile	Districe	Dishler	Like	Dislike	3/30/2021 19:09:14	3/30/2021 19:09:14	3/30/2021 19:15:42	3/30/2021 19:15:42	3/30/2021 20:27:00	3/30/2021 20:27:00	3/31/2021 11:14:22	3/31/2021 11:14:22
	Like	Disting	Lifer	Like	Distilice		Distile		Distiler			Like			3/31/2021 13:01:20	3/31/2021 13:01:20				
3/31/2021 14:19:59 Pop. Rock, Classical	Like		Like	Dislike	Like	Like	Dislike	Distike	Distiler	Dichlor		Like	5/9/2021 23:28:42	5/9/2021 23:28:42			3/31/2021 11:29:15		5/17/2021 12:54:35	5/17/2021 12:54:35
3/31/2021 15:45:07 Classical	Dislike	Distilor	Dichlor	Dislike	Distilor	Dishlor	Dislike	Distrike	Distiler	Dieblee	Like	Didike	3/31/2021 16:55:03	3/31/2021 16:55:03				3/31/2021 11:29:15	3/31/2021 23:14:02	3/31/2021 23:14:02
3/31/2021 17:19:58 Pop. Bock	Like	Distilor	Diehlor	Dislike	Like	Dishko	Didike	Didike	Distiller	Dichlo	Like	Like			3/31/2021 17:45:21	3/31/2021 17:45:21	3/30/2021 19:08:20	3/30/2021 19:08:20	5/10/2021 7:53:24	5/10/2021 7:53:24
3/31/2021 17:34:50 Pop. Country/Folk, Rap/Hip Hop. Classical	Like	Distiler	Like	Like	Like	Like	Like	Didtke	Distiller	Dichlo	Like	Diddle	3/30/2021 20:49:18	3/30/2021 20:49:18	4/4/2021 12:39:59	4/4/2021 12:39:59	4/3/2021 15:43:11	4/3/2021 15:43:11	4/3/2021 11:18:50	4/3/2021 11:18:50
3/31/2021 23:14:02 Pop. Bap/Hip Hop	Like	Like	Like	Distike	Like	Like	Like	Diddle	Didde	Like	Like	Like					3/31/2021 11:29:15	3/31/2021 11:29:15	3/31/2021 15:45:07	3/31/2021 15:45:07
4/1/2021 10:59:53 Pop. Bock, Classical	Like	Like	Like	Distilier	Like	Didle	Dishler	Diddle	Dishler	Dishler	Like	Like	4/1/2021 10:53:08	4/1/2021 10:53:08		1				
4/3/2021 11:18:50 Pop. Jazz/Blues, Rock, Rap/Hip Hop	Like	Dishler	Liles	Distilie	Like	Like	Like	Distilie	Distilie	Like	Like	Like	7,7,200 10000	.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			5/10/2021 7:28:45	5/10/2021 7:28:45		
4/3/2021 11:18:50 Pop. 3822/1880s, Bock, Rap/mp Bop 4/3/2021 15:43:11 Pop.	Like	Distilia	Diables	Distille	Libr	Dishler	Like	Distiller	Dishilor	Like		Distile					15 115 AUGUST 1 AUGUST	10 10 10 10 10 10 10 10 10 10 10 10 10 1		_
4/3/2021 15:43:11 Pop 4/5/2021 21:37:17 Pop. Bock. Country/Folk	Like	Distriction	Dichke Dichke	Tile	Like	Dishke Dishke	Turnet.	Like	Distriction	Like Dichler		Distike Distike				1			3/30/2021 20:49:59	3/30/2021 20:40:50
4/5/2021 2137:17 Pop. Rock, Collitry/Posk 4/7/2021 14:44:31 Pop. Jazz/Eliass, Rap/Hip Hop, Classical	Like	LOUISING TO SERVICE STATE OF THE PERSON STATE	LOCAL CO.	Distike	Later Co.	Distilo	ACCURAGE CO.	Distilie	Distilier	A CONTRACT		Like					3/30/2021 18:25:03	3/30/2021 18:25:03	5/10/2021 21:19:23	5/10/2021 21:19:23
4/1/2021 14:44:31 Pop. Jazz/neses, Rap/Hip Hop, Classical		1.160	L000		L160		District		Distriction	1,100						-	3/30/2021 18:25:03	3/30/2021 18:25:03	3/10/2021 21:19:23	3/10/2021 21:19:23
4/8/2021 21:51:49 Pop. Rock, Rap/Hip Hop	Like	Like	Dichlor	Like	Like	Dishlor	Like	Like	Distiller	Liler		Didike								
4/9/2021 0:23:57 Pop. Rock	Like			Dislike	Like	Didde	Dislike	Distike	Like	Dichler		Didike	4/9/2021 0:55:09	4/9/2021 0:55:09	4/9/2021 10:10:21			4/9/2021 16:03:19	4/9/2021 0:42:24	4/9/2021 0:42:24
4/9/2021 0:42:24 Jazz/Blues, Rock, Country/Folk, Classical	Like			Dislike	Like	Dishlor	Dishke		Districe			Distrike					4/9/2021 0:23:57	4/9/2021 0:23:57	1	
4/9/2021 16:03:19 Pop. Jazz/Blues, Rock, Country/Folk, Rap/Hip Hop. Classical	Like	Like	Like	Like	Like	Dishke	Like	Like	Like	Like	Like	Like								
4/11/2021 18:49:19 Pop. Country/Folk, Rap/Hip Hop, Classical	Like	Distiler	Lifer	Distike	Distilor	Dishke	Like	Like	Didde	Dichlo	Like	Like	4/9/2021 20:48:44	4/9/2021 20:48:44	4/27/2021 21:17:28	4/27/2021 21:17:28	3/30/2021 19:16:50	3/30/2021 19:16:50	3/30/2021 19:08:20	3/30/2021 19:08:20
4/12/2021 16:24:40 Pop. Jazz/Bluov, Country/Folk	Like	Dishler	Like	Like	Like	Distilor	Dishler	Distile	Dishler	Dishler	Like	Distiller					3/31/2021 11:36:00	3/31/2021 11:36:00	5/17/2021 12:54:35	5/17/2021 12:54:35
4/13/2021 20:33:15 Pop	Like	Dishler	Like	Dishke	Like	Dishler	Like	Dishike	Dishler	Dishler	Liles	Distile					3,31,232 113333	3,31,232 113333	3,11,232	3,11,2321123133
4/15/2021 16:35:21 Pop. Rap/Hip Hop. Classical	Like	Distilier	Dielike	Dislike	Like	Distilor	Distiller	Distike	Division	Like	Distilier	Distike							4/3/2021 11:18:50	4/3/2021 11:18:50
4/17/2021 20:02:48 Pop	Like		Dishlor	Dislike	Disdice		Like	Like	Division	Dishke		Distilier							0.0000000000000000000000000000000000000	0.00 4044 4404000
4/18/2021 13:16:53 Pop. Classical	Like	Libr	Dichlor	Dislike	Libra		Like	Distiles	Distinc	Dishler		Distile			4/18/2021 13:17:33	4/18/2021 13:17:33	3/30/2021 19:08:20	3/30/2021 19:08:20	5/17/2021 12:54:35	5/17/2021 12:54:35
4/18/2021 20:17:28 Pop. Country/Folk, Classical	1100	1.100	DOMEST.	Like	Dishler	Didlo	Like	Distilie	7.5	T.S.		Distile	4/20/2021 22:22:37		4/18/2021 20:26:35			4/20/2021 22:25:50		4/18/2021 20:37:13
	Like	1.160	Like	Dialike	Distinct	Diable	Like		Dishler	Like		Like					3/30/2021 22:25:90			3/30/2021 19:08:20
4/19/2021 11:25:16 Classical		Disting			Like			Like	Districe				4/19/2021 14:29:22	4/19/2021 14:29:22	5/11/2021 17:10:22	5/11/2021 17:10:22	3/30/2021 18:25:03	3/30/2021 18:25:03	3/30/2021 19:08:20	3/30/2021 19:08:20
4/19/2021 15:54:41 Pop. Jazz/Blues, Rock, Country/Folk, Rap/Hip Hop. Classical	Like	Like	Like	Like	Like	Like	Like	Like	Like	Like		Like								
4/20/2021 22:22:37 Pop. Rap/Hip Hop	Like	Like	Dishke	Dislike	Dishler		Like		Like	Like		Distrike	4/18/2021 20:17:28		4/18/2021 20:26:35	4/18/2021 20:26:35	4/18/2021 20:37:13	4/18/2021 20:37:13	4/20/2021 22:25:50	4/20/2021 22:25:50
4/23/2021 11:48:10 Pop. Jazz/Blues, Rock	Like	Distilor	Like	Like	Like	Dishke	Like	Didtke	Distiller	Like	Like	Distilie	5/11/2021 20:14:42	5/11/2021 20:14:42			5/12/2021 11:04:26	5/12/2021 11:04:26		
4/25/2021 14:12:21 Pop. Jazz/Elnes, Rock, Country/Folk, Rap/Hip Hop. Classical	Like	Like	Lifer	Like	Like	Dishke	Like	Like	Distiller	Like	Like	Like					4/8/2021 21:51:49	4/8/2021 21:51:49		
479679091 18-07-40 Pop. Book. Ran/Hin Hop.	Dislike	Dishler	Dishler	Dislike	Dishler	Didlo	Like	Distile	Like	Like	Like	Distile								
4/26/2021 18:07:40 Pop. Rock, Rap/Hip Hop 4/27/2021 20:09:21 Pop. Jazz/Stave, Cassical	Like	Like	Dishke Like	Distike	Like	Dishke	Dislike	Like	Distilion	Dielle	Like	Like	5/11/2021 15:29:29	5/11/2021 15:29:29	4/28/2021 19:58:16	4/28/2021 19:58:16	3/30/2021 18:25:03	3/30/2021 18:25:03	5/17/2021 12:54:35	5/17/2021 12:54:35
5/4/2021 14:03:26 Pop. Country/Folk	Like	Like	Dishler	Dislike	Like	Didlo	Like	Dislike	Dischier	Liles	Like	Didike								
5/4/2021 23:40:11 Pop. Jazz/Elnes, Rock, Country/Folk, Rap/Hip Hop. Classical	Libe	Like	Like	Like	Like	Like	Dishler	Like	Like	Like		Like								
5/5/2021 15/05/22 Pop. Jazz/Blues, Rock, Classical	Like		Like	Like	Like	Didlo	Distile	Distile	Dishler			Like	5/5/2021 15:12:23	5/5/2021 15:12:23						
5/5/2021 16:46:09 Pop. Ban/Hin Hop. Classical	Like		Dieble	Distiller	1.0		Like	Distile	Library.			Like	0/0/2021 10:12:20	3/3/2021 19:12:23						
5/5/2021 912:35 Pop. Bap/Hip Hop. Classical	LIBS	Distinct	LUCIAN	LUCIEC	L160	1,000	Like	L/DITE:	Liber	1,1300										
	Line	179/100	L100	LYCKER	LYDEROS	LYDGEO:	LIBO	LINETTED	L300	1,100	Like	Distribe	5/6/2021 9:16:58	5/6/2021 9:16:58			3/31/2021 23:14:02	3/31/2021 23:14:02		
5/9/2021 22:02:58 Pop. Rap/Hip Hop	Like	Like	Dishlor	Dislike	Like	Dudike	Like	Distille	Like	Dishke	Like	Dislike	5/9/2021 22:04:19	5/9/2021 22:04:19	5/9/2021 22:05:54	5/9/2021 22:05:54				
5/9/2021 22:22:38 Pop. Rock, Country/Folk, Classical	Like	Like	Like	Dislike	Dishler	Like	Dislike	Like	Like	Dishke		Like					5/10/2021 21:19:23	5/10/2021 21:19:23	4/7/2021 14:44:31	4/7/2021 14:44:31
5/9/2021 22:45:20 Pop. Rap/Hip Hop	Like	Like	Dichlor	Dislike	Dishlor	Dishke	Like	Distrike	Distrike	Like	Dislike	Dislike								
5/9/2021 22:47:55 Pop. Jazz/Blues, Rock, Rap/Hip Hop. Classical	Like	Like	Lifer	Dislike	Like	Lifer	Like	Distitor	Distile	Diello	Like	Dislike								
5/10/2021 0:22:24 Jazz/Blues, Classical	Dislike	Dislike	Lifer	Like	Like	Dishke	Like	Distibe	Distile	Like	Dislike	Like					3/31/2021 1:00:09	3/31/2021 1:00:00		
5/10/2021 7:27:37 Pop. Jazz/Bluov, Rock, Country/Folk, Classical	Like	Like	Lifer	Like	Like	Life	Like	Distilie	Didde	Diebler	Like	Like								
5/10/2021 7:28:45 Pop	Dishike	Like	Dishlor	Dislike	Dishler	Distilo	Like	Distile	Didde	Dishke	Like	Didike					4/3/2021 11:18:50	4/3/2021 11:18:50	t	_
5/10/2021 7:53:24 Pop. Bock, Country/Folk	Like	Like	Like	Like	Like	Dishler	Like	Dishike	Like	Like	Like	Distiller							3/31/3021 17:19:58	3/31/3021 17:19:58
5/10/2021 16:33:10 Pop. Jazz/Blue, Claudcal	Like		Life	Uke	Like	Distilor	Distiles	Distile	District	Like		Diddle	5/10/2021 18:38:21	5/10/2021 18:38:21			3/31/2021 11:29:15	2.231.2001.11.20.13	5/17/2021 15:59-27	5/17/2021 15:59:27
5/10/2021 20:38:15 Pop. Jazz/Blass, Cassacia 5/10/2021 20:38:15 Pop. Jazz/Blass, Bock, Country/Folk, Rap/Hip Hop	Like		Like	Like	Later Co.	Distilo	Like	Like	Dishler	Dieble		Didike	of the work to accept	syroyanar re-se-21		1	JyJ4/4041 11:29:15	3534/4044 1E2915	37 117 MAIN 10039521	97.117.494.1 195395.21
5/10/2021 20:38:15 Pop. Jazz/Blues, Hock, Country/Felk, Rap/Hip Hop 5/10/2021 21:19:23 Pop. Bock, Rap/Hip Hop	Like	LISTED	Like Dishlor	Like Dishke	L160	Dishke		Like	Lineson	Dichler		Distilie					5/9/2021 22:22:38			
		Like			Like	Distilor	Like		Lifee	Lifer	Distike	Didike					5/9/2021 22:22:38	5/9/2021 22:22:38		
5/11/2021 5:07:23 Pop. Rock, Rap/Hip Hop. Classical	Dislike	Dielike	Like	Like	Like	Distilor	Distike	Distille	Lifer	Like	Like	Distike							1	
5/11/2021 14:55:18 Jazz/Blues, Rap/Hip Hop, Classical	Like		Lifer	Like	Distilor	Didlo	Like	Dislike	Like	Lifer	Like	Like								
5/12/2021 11:04:26 Pop. Rock, Rap/Hip Hop	Like	Like		Like	Like	Like	Dislike	Dislike	Distiller	Like		Distilie								
5/12/2021 22:07:21 Pop. Jazz/Bluos, Rock, Classical	Like	Distiler	Like	Dislike	Like	Dishlor	Like	Distrike	Like	Dielike	Like	Distike					5/10/2021 0:22:24	5/10/2021 0:22:24	3/31/2021 1:00:09	3/31/2021 1:00:09
5/13/2021 14:11:30 Pop. Jazz/Blues, Country/Folk, Classical	Dislike	Distilion	Lifer	Dislike	Dishlee	Dishke	Dislike	Distile	Distiler	Dielike	Like	Distike								
5/17/2021 12:43:58 Pop. Jazz/Blues, Rock, Rap/Hip Hop	Like	Didice	Dieblee	Distike	Like	Didlo	Like	Diddle	Didde	Like	Distike	Like								
5/17/2021 12:54:35 Pop. Classical	Distiller	Distiles	District	Distilia	Dishler	District	Dishler	Distiller	Dishler	Dishler	Like	100								
5/17/2021 13:22:20 Pos. Jazz/Blue, Classical	Distile	Like	Like	Uke	Like	Dishler	Distiller	Distile	Distilier	Dishler	Dile	Distiller				1	3/31/2021 11:36:00	3/31/2021 11:36:00		
5/17/2021 15:59:27 Pop. Rock, Country/Folk, Classical	Distile	Like	Life	Distilier	Like	Like	Dichte	Distilie	Distilie	Dishler	Like	Like					3,33,434, 31,36,56	3,33,434, 31,39,09	3/31/3021 23:14:02	3/31/2021 23:14:02
3/31/2021 11:14:22 Pop. Book, Country/Posk, Casoscia	Like	1.00	Like	Distille	F-10	Dishler	Distille	Distille	Distance Control	Dishler	Diable	Distile					3/31/2021 12-45-53	3/31/2021 12-45-53	3/31/2021 23:14302	3/31/2021 23:14392
3/31/2021 11:14:22 Pop. Rock 3/30/2021 20:27:00 Pop. Classical	Like	Land E.O.	Like	Distike	Diables			Distile	Energia Company			Distile				1		3/31/2021 12:45:53		3/30/2021 20:27:00 3/31/2021 11:14:22
a/30/2021 2021:00 Pop. Classical	LIE	1.100	Like Diables	Didike	Litudifor	Like Dishler	Like		Linetiles	Like	Like	Didike Didike				3/27/2021 23:16:19	3/31/2021 12:45:53 3/30/2021 19:08:20			
3/27/2021 13:30:39 Pop	Line	1.100			TABILITIES.		LIBO	Like	District.	1,100	Literate		3/27/2021 14:24:36	3/27/2021 14:24:36	3/27/2021 23:16:19		3/30/2021 19:08:20	3/30/2021 19:08:20	5/17/2021 12:54:35	5/17/2021 12:54:35
3/31/2021 11:3k:00 Pop. Rap/Hip Hop	Like	Like	Dichlor	Dislike	Distiller	Didlo	Like	Dislike	Distiller	Like		Didike	3/31/2021 13:59:28	3/31/2021 13:59:28	3/31/2021 11:29:15	3/31/2021 11:29:15				
4/18/2021 20:37:13 Pop. Jazz/Blues, Country/Folk	Dislike			Like	Distilion	Dishlor	Dishke	Like	Distiler	Diebler		Dislike								
4/20/2021 22:25:50 Pop. Jazz/Blues, Rock, Country/Folk, Rap/Hip Hop, Classical	Like	Like	Like	Like	Distilor	Dishlor	Like	Like	Like	Like	Like	Like								

Timestamp	Check all of the following which you enjoy listening to:	Music #1 Opinion	Music #2 Opinion	Music #3 Opinion	Music #4 Opinion	Music #5 Opinion	Music #6 Opinion	Music #7 Opinion	Music #8 Opinion	Music #9 Opinion	Music #10 Opinion	Music #11 Opinion	Music #12 Opinion
3/30/2021 19:09:14		Like	Like	Dislike	Dislike		Dislike	Like	Dislike	Like	Like	Like	Dislike
	Jazz/Blues, Rock, Country/Folk, Classical	Dislike	Dislike	Like	Like		Dislike	Dislike	Like	Dislike	Dislike	Like	Dislike
	Pop. Rock, Country/Folk		Dislike	Dislike	Dislike		Dislike	Like	Dislike	Dislike	Dislike	Like	Dislike
	Pop. Jazz/Blues, Rock, Country/Folk, Rap/Hip Hop		Like	Like	Dislike				Dislike			Like	Dislike
3/30/2021 19:48:57			Dislike	Like	Like		Dislike	Dislike	Like	Dislike	Dislike	Like	Like
3/30/2021 19:45:51		Dislike	Like	Dislike	Dislike		Like		Dislike	Like	Like	Dislike	Dislike
3/30/2021 20:44:13			Like	Dislike	Dislike		Dislike		Dislike	Dislike	Like	Dislike	Dislike
	Pop, Rap/Hip Hop, Classical		Dislike	Dislike	Dislike				Dislike				Dislike
	Pop, Rock, Country/Folk, Rap/Hip Hop		Like	Like	Like		Dislike	Like	Dislike	Dislike	Like	Like	Dislike
3/30/2021 21:56:23			Dislike	Like	Like		Like	Like	Like	Dislike	Like	Like	Like
			Like	Dislike	Dislike				Dislike	Dislike	Like	Dislike	Dislike
3/31/2021 11:36:00			Dislike	Like	Dislike			Like	Like	Dislike		Like	Like
	Rock, Country/Folk, Classical												
	Pop, Country/Folk, Rap/Hip Hop	Like Dislike	Like Like	Like Dislike	Dislike Dislike	Dislike Like	Dislike Like	Like Dislike	Dislike Dislike	Like Like	Like Dislike	Dislike Dislike	Dislike Dislike
	Rock, Rap/Hip Hop												
	Pop, Jazz/Blues, Rock, Rap/Hip Hop, Classical		Dislike	Like	Dislike				Like		Like	Like	Dislike
	Pop, Rock, Country/Folk, Rap/Hip Hop, Classical		Dislike	Like	Dislike		Dislike		Dislike	Like	Like	Like	Like
	Pop, Jazz/Blues, Rock		Dislike	Like	Like		Like	Dislike	Like	Dislike	Dislike	Dislike	Like
	Pop, Rock, Country/Folk, Classical		Dislike	Like	Like		Dislike	Dislike	Like	Dislike	Dislike	Like	Like
3/31/2021 22:32:00	Jazz/Blues, Rock, Classical		Dislike	Like	Like		Like	Dislike	Dislike	Dislike	Like	Like	Like
4/1/2021 10:53:08	Rock, Classical		Dislike	Like	Dislike		Dislike	Dislike	Dislike	Dislike	Dislike	Like	Dislike
4/4/2021 12:39:59	Country/Folk, Classical		Like	Like	Dislike		Like	Dislike	Like	Dislike	Dislike	Like	Like
4/9/2021 0:55:09	Pop, Classical	Dislike	Like	Dislike	Dislike		Like	Dislike	Dislike	Dislike	Dislike	Like	Like
4/9/2021 10:10:21	Pop, Jazz/Blues, Rock, Classical		Dislike	Like	Like		Like	Dislike	Dislike	Dislike	Like	Like	Like
	Pop, Rock, Country/Folk, Rap/Hip Hop		Dislike	Dislike	Dislike		Dislike		Dislike	Dislike	Like	Dislike	Dislike
	Pop, Jazz/Blues, Country/Folk, Classical		Dislike	Like	Dislike		Dislike	Dislike	Like	Dislike	Dislike	Like	Dislike
4/18/2021 13:17:33		Like	Like	Dislike	Dislike		Like	Like	Like	Dislike	Dislike	Dislike	Like
4/18/2021 20:26:35		Dislike	Like	Like	Like		Dislike	Like	Like	Dislike	Dislike	Like	Dislike
	Pop, Jazz/Blues, Rock, Country/Folk, Classical		Dislike	Like	Like		Dislike	Dislike	Like	Dislike	Dislike	Like	Dislike
	Pop, Country/Folk, Classical		Dislike	Like	Dislike				Dislike				Dislike
4/28/2021 19:58:16	Jazz/Blues, Rock, Classical	Dislike	Dislike	Like	Dislike	Like	Dislike	Dislike	Dislike	Dislike	Dislike	Like	Dislike
	Pop, Country/Folk, Classical		Dislike	Like	Dislike		Dislike	Dislike	Dislike	Dislike	Like	Like	Dislike
	Rock, Country/Folk, Classical		Dislike	Like	Like			Dislike	Like	Dislike		Like	Dislike
	Pop, Country/Folk		Like	Like	Like		Dislike		Dislike			Like	Dislike
5/5/2021 23:40:43	Pop, Jazz/Blues, Rock, Rap/Hip Hop, Classical		Dislike	Like	Like		Dislike	Dislike	Dislike	Like	Like	Like	Like
5/6/2021 9:16:58	Pop, Classical		Dislike	Like	Dislike		Dislike	Like	Dislike	Dislike	Dislike	Like	Dislike
	Pop, Country/Folk, Classical		Dislike	Like	Dislike		Dislike	Dislike	Dislike	Dislike	Dislike	Dislike	Dislike
	Pop, Country/Folk		Dislike	Like	Like				Dislike	Like		Like	Dislike
5/9/2021 22:40:12	Classical		Dislike	Dislike	Dislike			Like	Dislike	Dislike	Dislike	Like	Dislike
5/9/2021 23:28:42			Dislike	Like	Dislike		Dislike		Dislike	Dislike	Dislike	Like	Dislike
	Rock, Classical		Dislike	Like	Like				Dislike	Dislike		Like	Dislike
	Rock, Country/Folk		Dislike	Like	Like				Like	Dislike		Like	Dislike
5/10/2021 12:22:06	Jazz/Blues, Rock, Classical		Dislike	Like	Like		Like	Dislike	Like	Dislike	Dislike	Like	Like
	Pop, Rock, Rap/Hip Hop, Classical		Like	Dislike	Like		Like	Like	Like	Like	Like	Dislike	Like
5/10/2021 8:32:01	Jazz/Blues		Like	Like	Like		Like	Like	Like	Like	Like	Like	Dislike
	Pop, Rock, Rap/Hip Hop		Dislike	Dislike	Dislike		Like	Like	Dislike	Like		Like	Dislike
	Pop, Rock, Country/Folk, Classical		Dislike	Like	Like			Dislike	Like	Dislike		Like	Dislike
	Pop, Jazz/Blues, Rock, Classical		Dislike	Like	Like		Dislike	Dislike	Dislike	Dislike	Like	Like	Dislike
	Jazz/Blues, Rock, Country/Folk, Rap/Hip Hop, Classical		Dislike	Like	Like				Dislike	Dislike	Like	Dislike	Like
5/10/2021 22:22:58	Jazz/Blues, Rock, Rap/Hip Hop, Classical		Like	Like	Like		Like		Dislike			Like	Dislike
5/10/2021 23:25:39	Jazz/Blues, Rock, Classical		Dislike	Like	Like		Like	Dislike	Dislike	Dislike	Dislike	Like	Like
	Pop, Rap/Hip Hop	Like	Dislike	Dislike	Like		Like	Like	Dislike	Like	Like	Dislike	Dislike
5/11/2021 11:41:56	Rap/Hip Hop, Classical	Dislike	Dislike	Like	Dislike		Dislike	Dislike	Dislike	Dislike	Like	Like	Dislike
5/11/2021 15:29:29	Classical	Like	Like	Like	Dislike		Dislike	Dislike	Dislike	Dislike	Dislike	Like	Like
5/11/2021 17:10:22	Jazz/Blues, Classical		Dislike	Like	Dislike		Dislike	Dislike	Dislike	Like	Dislike	Like	Dislike
	Rock, Rap/Hip Hop		Dislike	Like	Dislike		Dislike	Dislike	Dislike	Dislike	Like	Dislike	Dislike
	Pop, Jazz/Blues, Rock, Country/Folk, Classical		Dislike	Like	Like		Dislike	Dislike	Like	Dislike	Dislike	Like	Dislike
			Like	Like	Like		Dislike	Like	Like	Dislike	Dislike	Like	Like
5/13/2021 10:04:34			Dislike	Dislike	Dislike		Dislike	Like	Dislike	Like	Dislike	Dislike	Dislike
	Jazz/Blues, Rock, Rap/Hip Hop, Classical	Like	Like	Dislike	Like		Like	Dislike	Dislike	Dislike	Like	Like	Dislike
	Pop, Jazz/Blues, Rock, Classical		Dislike	Like	Like		Dislike	Dislike	Dislike	Dislike	Like	Like	Dislike
	Pop, Jazz/Blues, Country/Folk		Dislike	Like	Like		Dislike	Dislike	Like	Dislike	Like	Like	Dislike
			Like	Like	Like			Like	Like	Like	Like	Like	Like
	Pop, Country/Folk, Classical	Like	Like	Dislike	Like			Like	Dislike	Like	Like	Like	Dislike
4/20/2021 22:22:37		Like	Like	Dislike	Dislike			Like	Dislike	Like		Dislike	Dislike
3/31/2021 11:29:15	Pop, Rock, Rap/Hip Hop	Like	Dislike	Like	Dislike	Like	Like	Dislike	Dislike	Dislike	Like	Dislike	Dislike