markov_rerun_parsed

January 26, 2021

1 Markov Chain feature

1.1 Idea

Here I'm rebooting the previous Markov model. Instead of using a rank transformation of the features within each album, I simply encoded the feature of track $i + 1$ as being higher or lower than the feature of track i.

With this idea, I was hoping to pick up on very simple ordering patterns (e.g. tracks becoming increasingly louder), which should reflect on transition matrix patterns. Also, this encoding of the data picks up on larger patterns of transitions, if compared to the rank-order attempt.

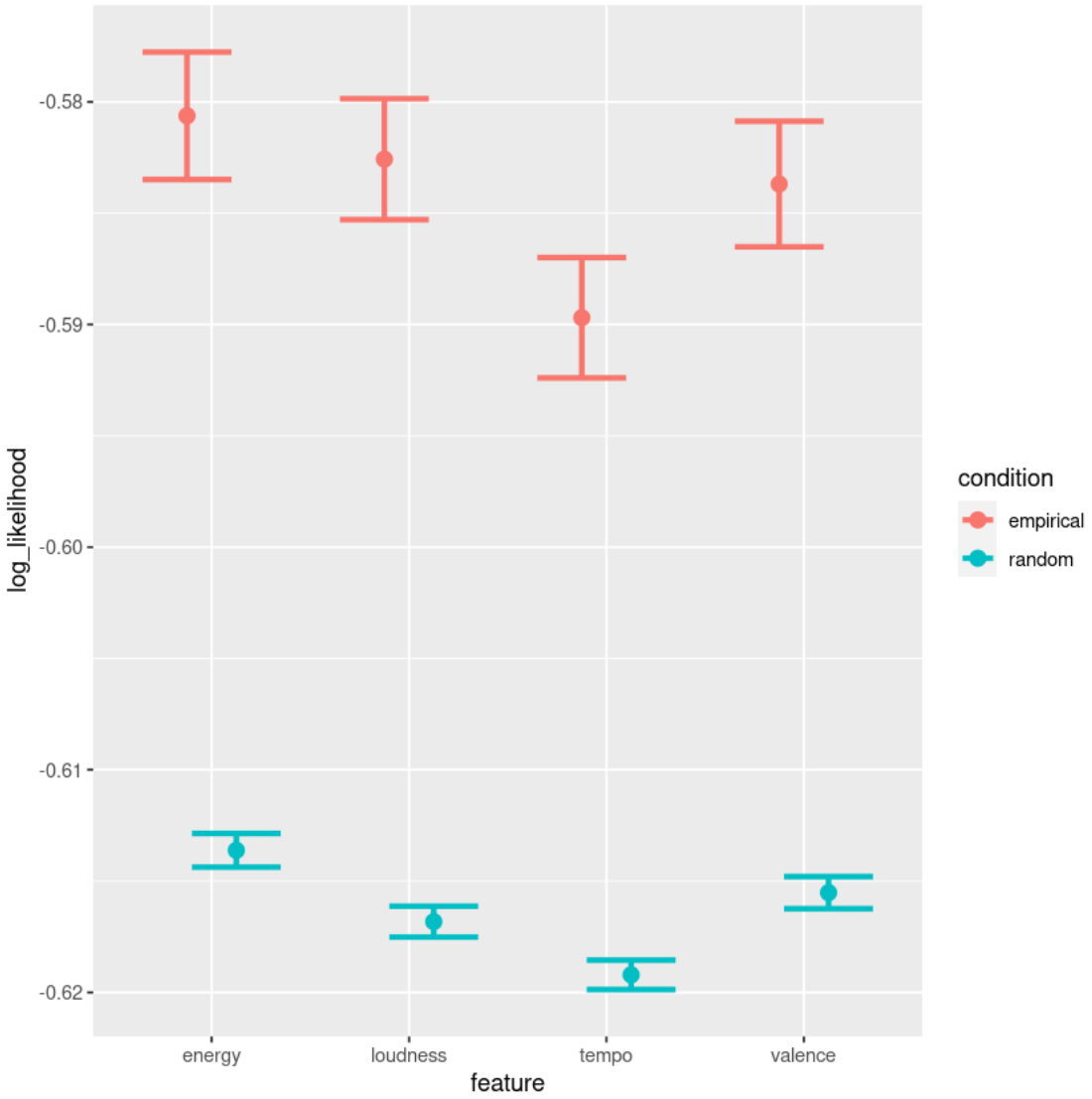
For instance, If the a feature of track i is set to "louder", it means that track $i - 1$ was "quieter" than track i. Therefore, the transition matrix from one state to the other essentially tells us the probability of a sequence of 3 tracks.

1.2 Evaluation

After encoding each feature transition, I kept two versions of the same training set: 1) the original one, and 2) a permutated vearsion of each album. These two data sets were used to fit two distinct sets of transition matrices (one for each feature, and one for each data set).

Within the training data set (20% of the albums), I evaluated the likelyhood of the data under two different markov models (the one fitted on the original albums and the one fitted on the permutated album).

Mean log likelyhoods were separatelly calculated for each album, and error bars represent a measure of dispersion across all albums.



2 Markov chains and track probability

2.1 Following idea

Even though the binary encoding of the data seems simple and naive, I think that we have some potential here, and this potential might relate to the interaction of transitions between each feature.

If take into account the transition probability of our 4 features, we should find, for instance, that the likelyhood of transitioning from track_number_1 to track_number_2 is generally the highest - or one of the highest, if compared to the other tracks within the album.

With that, we should be able to narrow down the best candidates for track $i + 1$, given the states of track i. I ilustrate this idea below.

The column named "track_2_candidates" indicates each track's likelyhood of appearing immediately after track 1. We see that tracks 2, 4 and 6 are the best candidates for following (with tied likelyhoods). The column named track_3_can says that, in fact, track 3 has the highest likelyhood of following track 2.

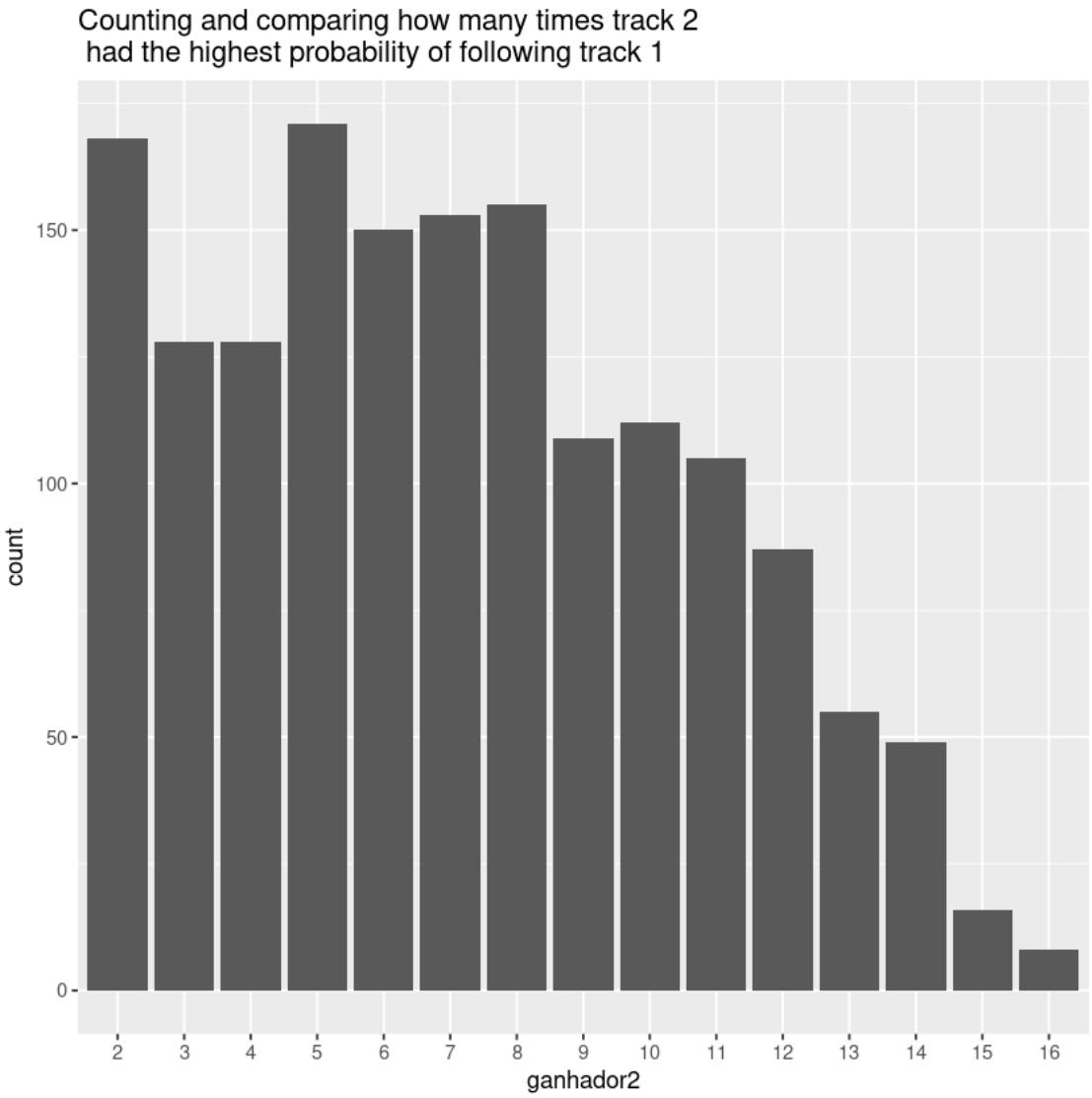
A tibble: 6 × 5

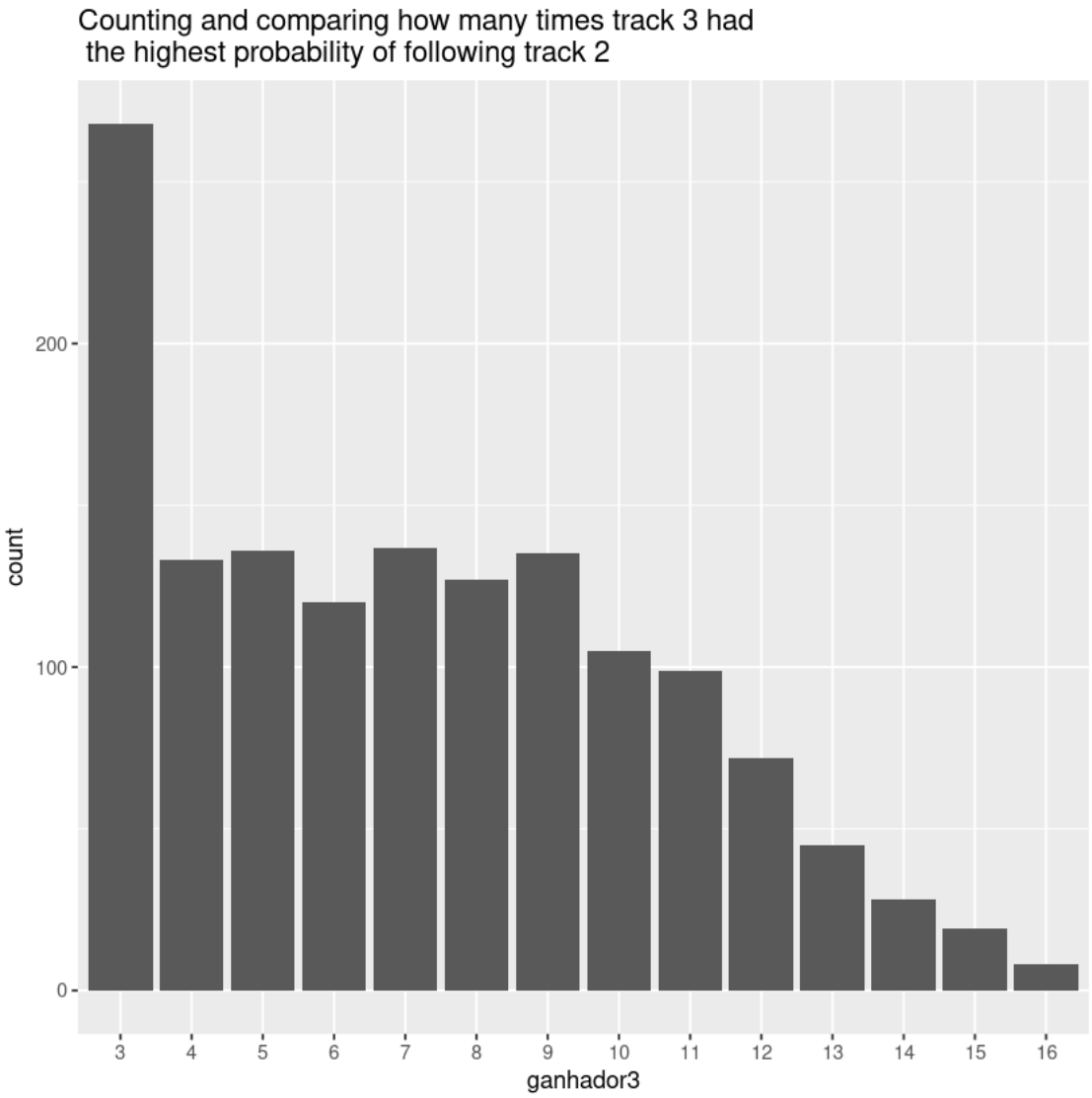
track_number <int>	valence <chr>	energy <chr>	track_2_candidates <dbl>	track_3_can <dbl>
1	start	start	0.00000000	0.00000000
2	greater	smaller	0.06345298	0.00000000
3	smaller	greater	0.06081012	0.19620454
4	greater	smaller	0.06345298	0.01248015
5	smaller	greater	0.06301077	0.09789498
6	greater	smaller	0.06345298	0.01248015

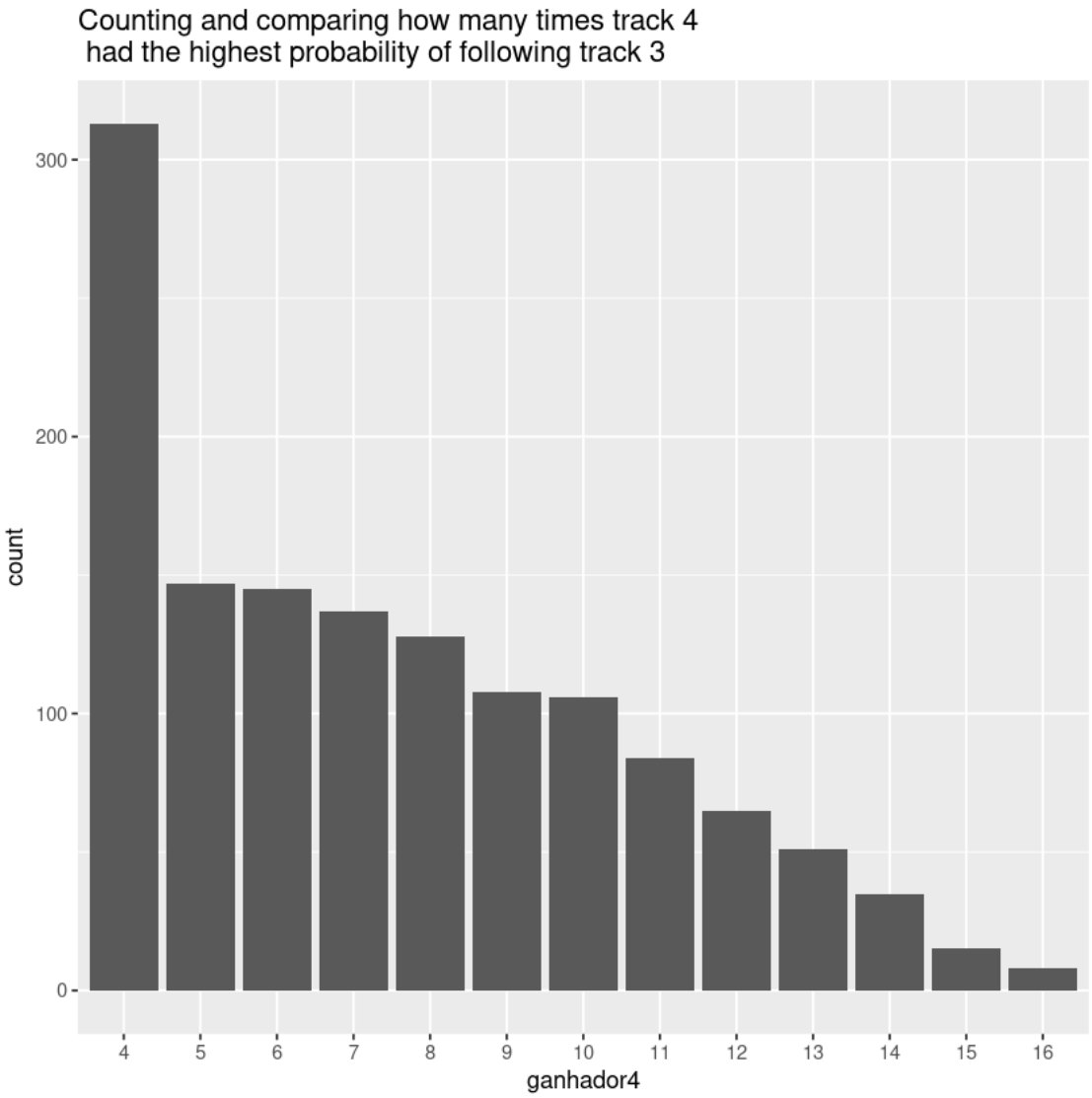
2.2 Generalizing to the whole test data set

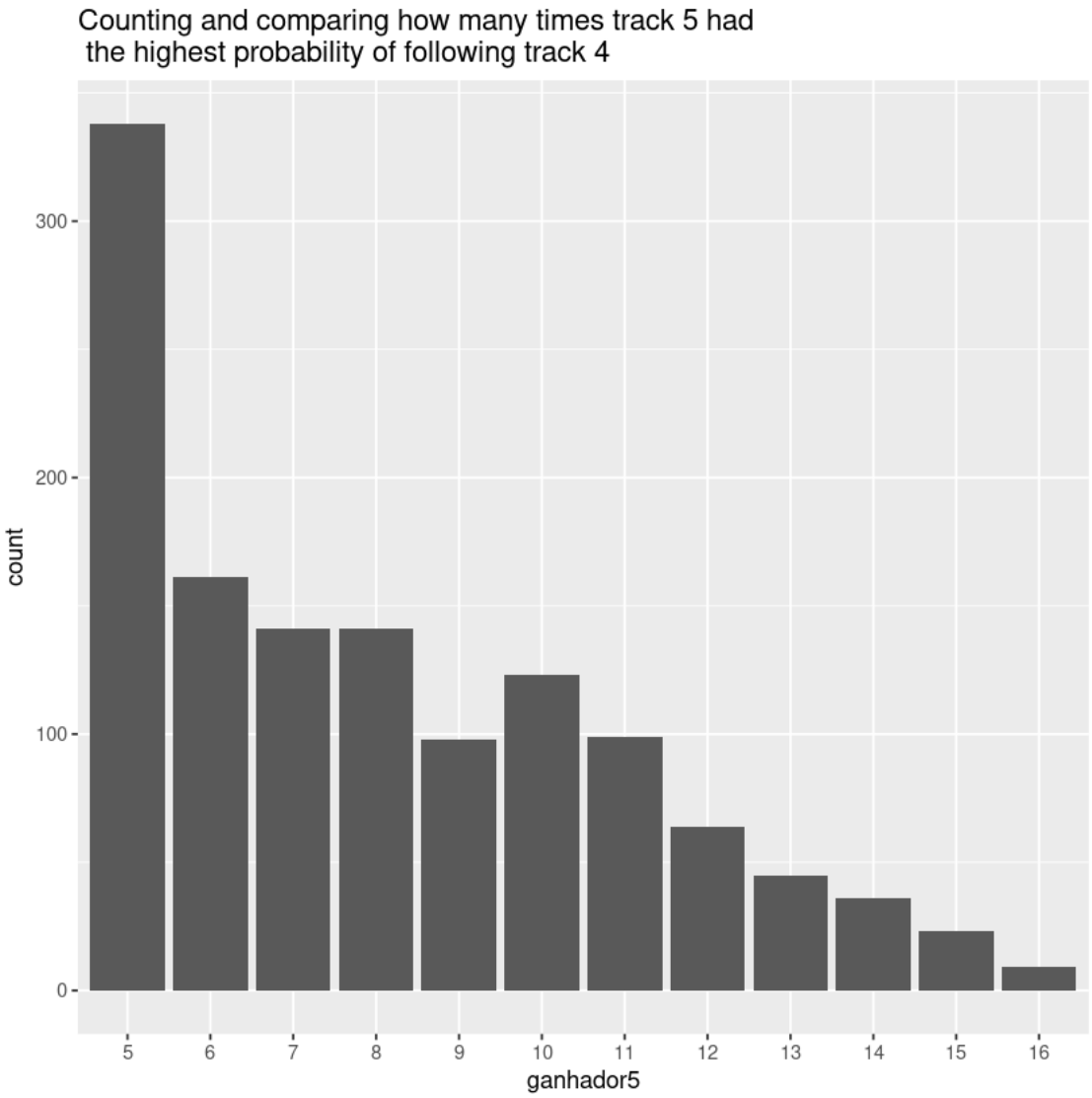
Now I'm veryfing the frequency with which track $i + 1$ is identifyed as the most likely track to follow track i. Results are displayed below.

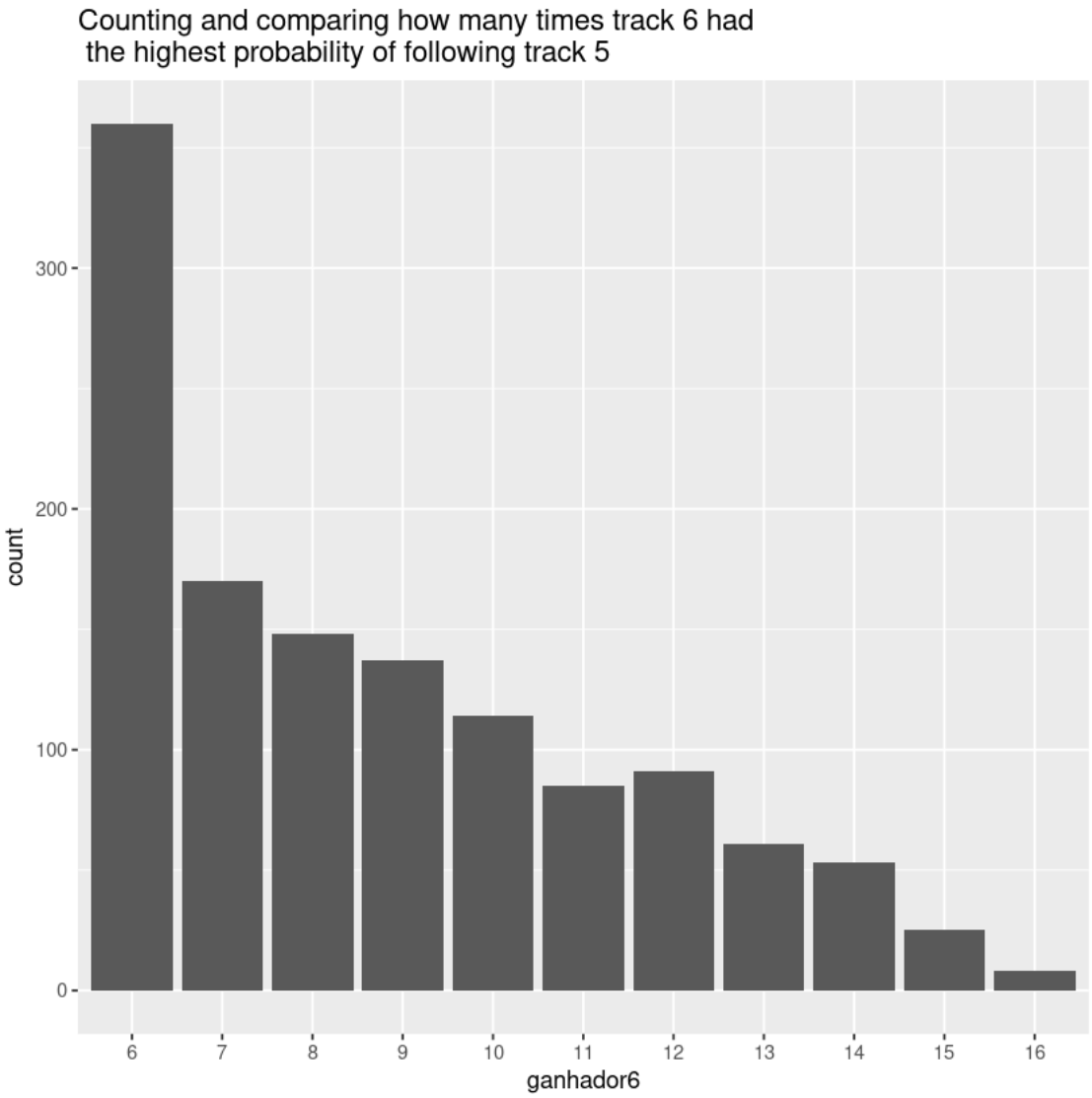
*Note: these are still exploratory and we have some limitations, such as the issue of imbalanced album lengths, as well as the fact that the first track of each album doesn't have a comparative feature.


Counting and comparing how many times track 2
had the highest probability of following track 1
count
150
100
50
0
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
ganhador2


Counting and comparing how many times track 3 had
the highest probability of following track 2
count
200
100
0
3
4
5
6
7
8
9
10
11
12
13
14
15
16
ganhador3


Counting and comparing how many times track 4
had the highest probability of following track 3
count
300
200
100
0
4
5
6
7
8
9
10
11
12
13
14
15
16
ganhador4


Counting and comparing how many times track 5 had
the highest probability of following track 4
count
300
200
100
0
5
6
7
8
9
10
11
12
13
14
15
16
ganhador5


Counting and comparing how many times track 6 had
the highest probability of following track 5
count
300
200
100
0
6
7
8
9
10
11
12
13
14
15
16
ganhador6