Annie Nguyen – STA502B

Dr. Steven Wright

I. Introduction

Original Statement for the Assigned Task :

Each movie corresponds to many different tags generated by different users. Write a SAS macro program which accepts a movie and outputs a list of key words that summarizes the different tags for that movie.

About the data

Six datasets in CSV format were provided for this project. These datasets are named genome-scores.csv, genome-tags.csv, tags.csv, movies.csv, links.csv, and ratings.csv. For the purpose of this project, only genome-scores.csv, genome-tags.csv, tags.csv, movies.csv are used.

Link to Google Drive for datasets and README file.

Project Description:

In this project, each movie is associated with multiple tags, contributed by various users. The objective is to develop a SAS macro program that takes a specific movie as input and generates a concise list of keywords. These keywords serve as a summary of the diverse tags attributed to the chosen movie, providing valuable insights into its content and user-generated metadata.

This SAS macro program streamlines the process of distilling extensive tag data into a condensed set of keywords, enhancing the accessibility and comprehensibility of information associated with each movie.

II. Tables and Graphs:

Table 1:

This table shows the first 14 movie with ID 1-14 associated with tagId = 1, tag ‘007’. The movie titles hasn’t been added for this table, so it could be somewhat difficult to understand.

List of movieIds and tags

Obs	movieId	tagId	tag
1	1	1	007
2	2	1	007
3	3	1	007
4	4	1	007
5	5	1	007
6	6	1	007
7	7	1	007
8	8	1	007
9	9	1	007
10	10	1	007
11	11	1	007
12	12	1	007
13	13	1	007
14	14	1	007

Table 2 :

This table is the edited version of table 1, with movie title added. It is sorted by movieId, then tagId.

List of movieIds and tags with titles

Obs	movieId	tagId	tag	title
1	1	1	007	Toy Story (1995)
2	1	2	007 (series)	Toy Story (1995)
3	1	3	18th century	Toy Story (1995)
4	1	4	1920s	Toy Story (1995)
5	1	5	1930s	Toy Story (1995)
6	1	6	1950s	Toy Story (1995)
7	1	7	1960s	Toy Story (1995)
8	1	8	1970s	Toy Story (1995)
9	1	9	1980s	Toy Story (1995)
10	1	10	19th century	Toy Story (1995)
11	1	11	3d	Toy Story (1995)
12	1	12	70mm	Toy Story (1995)
13	1	13	80s	Toy Story (1995)
14	1	14	9/11	Toy Story (1995)
15	1	15	aardman	Toy Story (1995)
16	1	16	aardman studios	Toy Story (1995)
17	1	17	abortion	Toy Story (1995)
18	1	18	absurd	Toy Story (1995)
19	1	19	action	Toy Story (1995)
20	1	20	action packed	Toy Story (1995)

Table 3:

This table shows the first 15 rows from the data set tags, containing tag, userId and movieId values. This table is sorted by movieId and to be merged with table 2 above to create a table of all tags with movie titles.

Tags for all movies

Obs	tag	userId	movieId
1	animated	1040	1
2	buddymovie	1040	1
3	Cartoon	1040	1
4	cgi	1040	1
5	comedy	1040	1
6	computeranimation	1040	1
7	family	1040	1
8	friendship	1040	1
9	kids	1040	1
10	toy	1040	1
11	toys	1040	1
12	adventure	5510	1
13	animated	5510	1
14	animation	5510	1
15	buddymovie	5510	1

Table 4:

This table shows the 15 most common tags for all movies, sorted by descending Frequency values from the merged table of table 2 and 3. This shows that the proc freq step works.

Most 15 common tags for all movies

The FREQ Procedure

tag	Frequency
sci-fi	9400
atmospheric	6430
action	6219
comedy	5923
surreal	5299
basedonabook	5296
funny	4864
twistending	4844
visuallyappealing	4333
dystopia	4268
darkcomedy	4027
BD-R	4024
romance	3867
fantasy	3829
stylized	3804
The first 15 levels are displayed.	

Table 5:

This table shows the 15 most common tags for Toy Story (1995), sorted by descending Frequency values. This shows that the macro works for customizable movie input.

List of 15 Most Common Keywords/Tags for Movie Toy Story (1995)

The FREQ Procedure

tag	Frequency
animation	80
Pixar	69
pixar	57
Disney	47
TomHanks	35
funny	35
children	32
computeranimation	32
toys	29
comedy	25
witty	24
family	23
friendship	22
adventure	18
animated	18
The first 15 levels are displayed.	

III. Final Documented SAS code:

```
/* Please change the folder path on the right-hand side of the statement
   below to the path where your data files are located.
*/
%let path = M:\STA502\Term Project\ml-latest;

/* Movie tags

This program reads the movie tags and titles from the movie data files
for a selected movie and generates a keyword summary for movies
based on tags' frequency.

The program is implemented as a SAS macro program. Please change
the statement

    %let path=

at the top of this file to specify the folder on the user's system
where all the data files are located. Next run this SAS file to
define the macro.

The first macro importFiles are to import the data set seamlessly.
It is invoked later on in the second macro movieTags_method, so no
further action apart from running this file is needed.

The second macro movieTags_method provides user flexibility,
is invoked as in this example:

    %movieTags_method(Toy Story (1995))

The program requires that the data files meet these
assumptions:

  1. All data files are CSV files, included in the movie data set:
        genome-scores.csv
            genome-tags.csv
            movies.csv
            tags.csv

  2. All data files are has to be in the same folder/path.

*/

/* Create a macro to read the CSV files */
%macro importFiles(in=, out=);
    proc import
        datafile=&in /* CSV data set */
        out=&out /* New SAS data set */
        dbms=csv
        replace; /* Replace existing dataset if it already exists */
        guessingrows=1000; /* Adjust the value as needed */
    run;
%mend importFiles;
```

```
/* Macro to generate movie tags/keywords */
%MACRO movieTags_method(moviename);

      /* Import CSV files using the macro */
      %importFiles(in="&path\genome-scores.csv", out=genome_scores);
      %importFiles(in="&path\genome-tags.csv", out=genome_tags);
      %importFiles(in="&path\movies.csv", out=movies);
      %importFiles(in="&path\tags.csv", out=tags);

      /* Sort the genome scores dataset by tagId */
      proc sort data=genome_scores out=genome_scores_sorted;
          by tagId;
      run;

      /* Format tag in genome_tags */
      data genome_tags;
            length tag $200;
            set genome_tags;
            format tag $200.; /* Apply a format to the tag variable */
            informat tag $200.;
            tag = compress(tag); /* Remove unnecessary spaces */
      run;

      /* Merge the sorted genome scores with genome tags */
      data genome_merged;
            merge genome_scores_sorted genome_tags;
            by tagId;
            drop relevance;
      run;

      /* Sort the genome_merged dataset by movieId */
      proc sort data=genome_merged out=genome_merged_sorted;
          by movieId;
      run;

      /* Sort the movies dataset by movieId */
      proc sort data=movies out=movies_sorted;
          by movieId;
      run;

      /* Merge genome_merged with movies */
      data movie_merged;
          merge genome_merged_sorted movies_sorted;
          by movieId;
            drop genres;
      run;

      /* Remove 'timestamp' variable from 'tags' dataset */
      data tags;
            length tag $200; /* Adjust a maximum length of 200 characters */
            set tags;
            tag = compress(tag); /* Remove unnecessary spaces */
            format tag $200.; /* Apply a format to the tag variable */
            informat tag $200.;
            drop timestamp;
            if cmiss(of _all_) then delete;
```

```
        run;

        /* Sort the 'tags' dataset by movieId */
        proc sort data=tags out=tags_sorted;
            by movieId;
        run;

        /* Merge 'movie_merged' with 'tags_sorted' */
        data movie_tagged;
            merge movie_merged tags_sorted;
            by movieId;
        run;

        /* Filter the dataset for the specified movie title */
        data movie_tags;
              length tag $200; /* Adjust a maximum length of 200 characters */
              set movie_tagged;
              tag = compress(tag); /* Remove unnecessary spaces */
              format tag $200.; /* Apply a format to the tag variable */
              informat tag $200.;
              where title = "&moviename";
        run;

        /* Generate Frequency Table:
        Create a frequency table for tags. Order the tags by frequency. */
        title "List of 15 Most Common Keywords/Tags for Movie &moviename";
        proc freq data=movie_tags order = FREQ;
            tables tag / out=tag_freq_movie nocum nopercent maxlevels=15;
        run;

%MEND movieTags_method;
```