Introduction to statistical analysis of phonetic data in R

Julián Villegas

University of Aizu, Japan
julian at u-aizu dot ac dot jp

Abstract

This manuscript summarizes the main points discussed during a workshop with the same title presented during the 2nd International Symposium on Applied Phonetics—ISAPh2018. Because of the time constraints of the workshop, this is a very limited introduction to R created with the intention of showing how to use R in the analysis of phonetic data and encourage attendees to use it in their research. Dataset for the examples, for a do-it-yourself part, and a tutorial are freely available from http://onkyo.u-aizu.ac.jp/classes/ez.

Keywords: R, Repeated Measures, ANOVA, ez library.

1. Motivation

Documenting techniques for teaching pronunciation (or related speech communication aspects) and comparing effects of such techniques are important to advance our current understanding of beneficial strategies in phonetics education. These processes, however, may be sometimes challenging because of unfamiliarity with appropriate tools for statistical analysis, plotting results, etc.

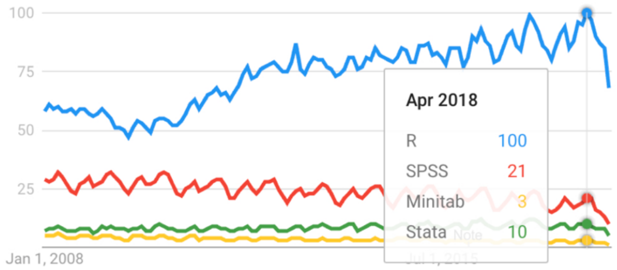


Figure 1: Relative interest over time for different statistics programming languages across the world. A value of 100 indicates a peak of popularity of the term as measured by Google Trends.

R, a rich environment for statistical analysis, can be considered as an underused tool in the field since despite its benefits some consider it difficult to learn. Nevertheless, in the last decade there has been a growing interest in R worldwide while interest in other traditional software for statistics declines, as shown in Figure 1 produced with Google Trends (https://tinyurl.com/ydg26k2w).

This worldwide trend is also reflected in a growing and vibrant community of users and developers that are in general generous with explanations and forgiving with novices. This community support eases the learning process for the interested researchers.

The objective of the workshop summarized here was to address common hurdles that researchers in phonetics may face in the process of data analysis using R. To that end, an example using Repeated Measures Analysis of Variance—RM-ANOVA is offered including methods for producing figures, tables, etc.

2. Installing R

Installing R is straightforward:

- Go to the Comprehensive R Archive Network (CRAN) website [1]
- Select a close mirror, e.g., https://cran.ism.ac.jp/ in Japan
- Select a precompiled binary distribution according to your operative system which will download an installer for the latest version of R (at writing time, R-3.5.1—“Feather Spray”).
- Open the installer and follow its instructions.

There is a popular (but optional) graphical user interface (GUI) for R called RStudio [2]. This article does not address how to use this GUI, but the code and libraries discussed here can be readily used in RStudio. Note however that R must be also installed in order to use RStudio.

R is capable of displaying GUI elements and messages in the language that is set as default by the operative system. For users of MacOS, the installation process may fail to correctly set this language. In consequence, this may cause problems with script outputs (e.g., being unable to correctly display non-alphanumeric characters in a figure). If that is the case, issuing the following command in R console solves that problem:

```
system("defaults write org.R-
project.R force.LANG en_US.UTF-8")
```

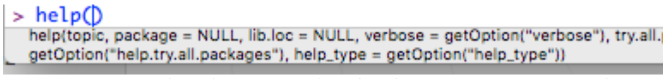
10.21437/ISAPh.2018-5

After issuing that command, R needs to be restarted. Note: Section 7 (instead of section 9, as suggested by a warning message) is the actual section of the MacOS FAQ where this issue is explained.

Installing and updating libraries is similarly easy by using the 'R Package Installer' from the Menu 'Packages & Data.' It is a good idea to let R install dependencies automatically and to periodically update the installed libraries.

3. Getting help

R has a comprehensive manual [3] (from which most of the content of this section is based upon) and built-in documentation that helps to understand how to use a function, operation, etc. To access such documentation, one can type in the console `help('command')` or `?'command'`; one can also select a written command with the mouse and simultaneously press the command+h keys to get the corresponding entry.



```
> help()
  help(topic, package = NULL, lib.loc = NULL, verbose = getOption("verbose"), try.all.
  getOption("help.try.all.packages"), help_type = getOption("help_type"))
```

Figure 2 Status bar documentation for the `help()` command.

Additionally, R has a useful feature to remind the arguments of a function (a.k.a command): When one type a command in the console, R shows the possible arguments and their defaults in the status bar as shown in Figure 2.

R has a function called `help.search('topic')` (and its alias `??`) which is useful to search for information on a given topic. For example, writing `??"comparison of proportions"` in the console will open a new tab (or window) of the default web browser with pointers to libraries relevant to the search topic if they exist. In the example above, one should get a hit (at least) suggesting to read the pairwise comparisons for proportions library entry (stats::pairwise.prop.test).

Other help sources include the command `example(topic)` which reproduces an example of the requested topic if there is one available; useful examples provided at the end of the help page of a topic; or manuals and vignettes available at [1] to learn more about a given library. A vignette in R parlance is a kind of tutorial accompanying a library.

There are numerous online resources and forums where to find information as well. Some of the most reputable ones are:

- https://stackoverflow.com/questions/tagged/r for R related questions,
- https://stats.stackexchange.com/ for statistics questions, and
- https://groups.google.com/forum/#!forum/ez4r for statistics questions, and for the ez library

When asking help on an Internet forum, always include the versions of your operative system, R, and relevant libraries as well as a working example if possible. For finding versions of R and libraries employed in a session, use the command `sessionInfo()`.

4. Terminology

Each language has its own jargon and R is not the exception. These are some of the most common terms with their meanings:

- **Vector**: An ordered collection of numbers (usually), e.g., `c(3,1,4,1,5,9)`.
- **Dataframe**: Dataframes are a collection of vectors in which the columns can be of different types. Usually, a row has one data observation with different aspects of the observation in different columns.
- **Factor**: A categorical variable in a dataframe may be considered a factor, and each of its categories a level.
- **Wide vs. long representation**: When extracting acoustic data from programs such as Praat [4], these data are often collected in so-called 'wide-format' spreadsheets and tables where there is a column for each variable, as shown in Table 1.

Table 1 A sample dataframe in wide-format representation

subjID	label	duration	f0	F1
subj1	con1	*0.07000*	*149.777*	*515.864*
subj1	voc1	*0.07616*	*144.938*	*591.371*
subj2	con1	*0.06765*	*137.835*	*327.424*
subj2	voc1	*0.09672*	*140.127*	*617.94*
subj3	con1	*0.07215*	*127.312*	*628.841*
subj3	voc1	*0.10034*	*126.121*	*526.501*

Although wide-format representation is convenient for computing statistics with some R functions and MS Excel (or other spreadsheet-based software), it presents disadvantages for many functions in R. For example, plotting and modeling functions in R require the data in long-format representation. In long-format representation, variable types are included in a column and their values in another. So, the same information presented in Table 1 can be transformed into a long-format representation as shown in Table 2.

The function `melt(data)` from the `data.table` library [5] can be used to convert between wide- and long-format. Alternatively, there are some programs such as VoiceSauce [6] that export data directly in long-format. In the end, the degree of 'wideness' of your data depends on your needs.

Table 2 The same data shown in **Table 1** A sample dataframe in wide-format representation**Table 1** in long-format representation.

subjID	label	type	value
subj1	con1	dur	*0.07000*
subj1	voc1	dur	*0.07616*
...			
subj1	con1	f0	*149.777*
subj1	voc1	f0	*144.938*
...			
subj1	con1	F1	*515.864*
subj1	voc1	F1	*591.371*
...			

5. Executing code

R is case sensitive: 'a' and 'A' are two different things. For naming variables, it is preferable to use only alphanumeric characters without accents, i.e., [A:Z,a:z,0:9,'_']. Variable names should start with a letter. Commands are separated by semicolons or a new line. The default prompt in the console is a '>' symbol. If a command is incomplete at the end of a line, R will change the '>' symbol to a '+' (by default). Vertical arrow keys can be used to scroll back and forth in the command history, and recall (and edit) previous issued commands. The [tab] key can be used to autocomplete commands, name of variables, etc. Autocomplete also works with some characters such as parentheses, curly braces, etc.

One can type in commands directly on the console for simple computations. But, when running statistical analyses, the number of lines may grow rapidly yielding this method rather cumbersome. For those instances, it is better to create a new document (command+n) and execute code from there.

Comments are an important piece of documentation either for others to understand a script or for future personal references. They are created by preceding text with a '#' symbol anywhere in the document. It is recommendable to have five sections on a script:

1. Start a document with a comment about the contents and purpose of the script, the author and creation date.
2. Load all the libraries you will need.
3. Insert all the functions you create.
4. Insert any other piece of code that does not change, and you may need (e.g., defining constants, options, etc.)
5. Insert the actual script code.

Items 1–4 may be run only once when you start an R session.

To execute all the contents of a file named 'scr,' one can issue the command `source('src')`. But, more frequently, one is interested in only running parts of the script. In that case, one should select with the mouse the desired lines of the script and press the keys command+return. Either way, be careful when re-running parts of a script as the contents of variables may change. Always read the output in the console and look for error and warning messages. Remember that just because R outputs a result does not mean that it is correct. Always be critical.

6. Step-by-step case

This section presents a case of study where repeated measures ANOVA has been used. The data corresponds to a study that investigates the effect of task on the intensity of speech in noisy conditions [7]. Data (in a file with values separated by commas) for this example is freely available from http://onkyo.u-aizu.ac.jp/classes/ez/.

Concretely, in that research we were interested in finding any differences in speech intensity of Japanese speakers subjected to alternating periods of silence and noise while engaged in four tasks that varied in their communication effort and purpose as shown in Table 3.

Table 3 Tasks of the experiment

	Communication effort	
	Inactive	Active
No goal	Soliloquy (S)	Free Dialog (D)
Goal	Text reading (T)	Battleship (G)

We wanted to know if speech levels in quiet and noisy conditions vary depending on whether the task requires an active communication effort, and whether the task is goal oriented.

Since we collected data from the same subjects under different conditions, we opted for analyzing these data using a within-subjects or Repeated Measures Analysis of Variance (RM-ANOVA). RM-ANOVA assumes equality of variance (often determined with a Mauchly's test). If not equal, degrees of freedom and F-ratios need to be adjusted either with Greenhouse-Geisser or Huynh-Feldt correction. An RM-ANOVA lets you remove variance due to between participant differences from the error term in the ANOVA. The effect of subjects is considered random. A full explanation about RM-ANOVA is beyond the scope of this article. Those interested in learning more about this analysis are referred to [8].

Data can be imported into R with this command:

```
LD=read.csv('data/LombardData.csv',
header=T)
```

The dataframe LD has several factors including:

- Speaker: subject IDs
- Time: time in s where the measure was taken
- Task: Soliloquy (S), Text reading (T), Free dialog (D), Battleship (G)
- Condition: Background condition (silence or noise)
- Goal: Has the task a planned goal (yes or no)?
- Comm: Has the task a planned communication effort (yes or no)?
- Intensity: speech level in dB

The RM-ANOVA is eased by using the `ezANOVA()` function of the `ez` library [9]. Some useful parameters of this function are:

- dv: the dependent variable (intensity)
- wid: the subjects (speaker)
- within: the within-subject factors considered in this analysis
- within_full: all the within-subject factors in the model
- between: the between-subject factors
- type: the type of sums of squares (set to 3 to have results as reported in SAS or SPSS)

The following command stores the RM_ANOVA results in the levAoV variable:

```
levAoV = ezANOVA (data=LD,
dv=.(intensity), wid=.(speaker),
within=.(condition, goal, comm),
type = 3)
```

Note how most ezANOVA() parameters need to be surrounded by `.()`. To read the results of this analysis one can use the function print:

```
print(levAoV)
```

This command generates a table that comprises the effect name, degrees of freedom in the numerator (DFn) and denominator (DFd), results of the F-statistics, p-values, whether they are significative at a confidence of 95%, and generalized eta-squared values (the size of the effect). There are many ways to report these results. For example, using the APA style, they should read somewhat like:

As summarized in Table 4, background condition, and communication effort were found to have a large effect [10] on speech intensity. Their interaction had a small but significant effect as well, along with the triple interaction between goal orientation, background condition, and communication effort. The effects of goal and other interactions were not significant.

Table 4 RM-ANOVA Results

Effect	Statistics	p-value	η^2_G
condition	*F(1,8)=151.28*	***<.001***	*0.249*
goal	*F(1,8)=1.929*	*0.202*	*0.020*
comm	*F(1,8)=30.131*	***<.001***	*0.394*
condition:goal	*F(1,8)=0.343*	*0.574*	*.000*
condition:comm	*F(1,8)=15.028*	***0.004***	*0.011*
goal:comm	*F(1,8)=0.591*	*0.464*	*0.006*
condition:goal:comm	*F(1,8)=7.797*	***0.023***	*0.004*

In general, after establishing the probabilities of having significant differences, one should run a post-hoc analysis to find out where these differences are located. The function `ezStats()` is useful to create a table with the descriptive analysis:

```
effectDescript = ezStats(data = LD,
dv=.(intensity), wid=.(speaker),
within=.(condition,goal,comm), type
= 3)
```

Note that the syntax for `ezStats()` is very similar to that of `ezANOVA()`. The resulting table shows the number of participants, means and standard deviations, and Fisher's Least Significant Difference (FLSD). The latter is a way to detect significant differences between levels: If the FLSD for two levels overlap, there is no significant difference between them. This is easier to see with a plot:

```
ezPlot(data=  LD,  dv=.(intensity),
wid=.(speaker),
within =.(condition,goal,comm),
x=condition,split = goal,col = comm)
```

The `ezPlot()` function also has a similar syntax as that of `ezANOVA()`, but adds some parameters to specify the way the plot is displayed, for example, `x` defines the abscissa, and `col` defines a given factor (i.e., comm) to create column panes for its levels (`row` works in a similar fashion); `split` lets you overlap two factors in the same pane. The resulting plot is shown in Figure 3. Although informative, this plot looks unappealing for several reasons:

- Gray background is sometimes inconvenient.
- Legend on the right takes space of the plot.
- Font size can be too small to include in a manuscript.

- Colors could also be problematic for publishing.

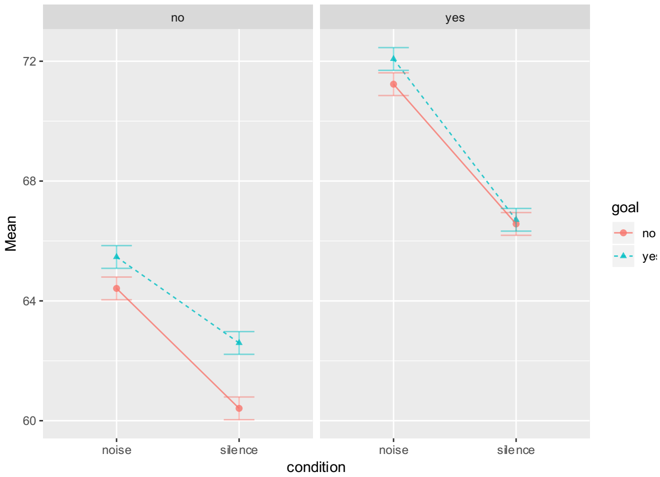


Figure 4 Plot generated with `ezPlot()`.

`ezPlot()` returns a kind of plot known as `ggplot()` from the library `ggplot2` [11]. We can address the aforementioned issues by capturing the output of `ezPlot()` in a variable and modifying it as shown in Figure 5. This process is not difficult, but somewhat long to be included in this article. Those interested on how to reformat an `ezPlot()` output are referred to the original material of the workshop available as ebook, PDF, or HTML at http://onkyo.u-aizu.ac.jp/introR/. In the same vein, a full description of the plotting possibilities brought by the `ggplot2` library is beyond the scope of this article, but the vignette of this library [11] is a very good starting point for those interested in learning more.

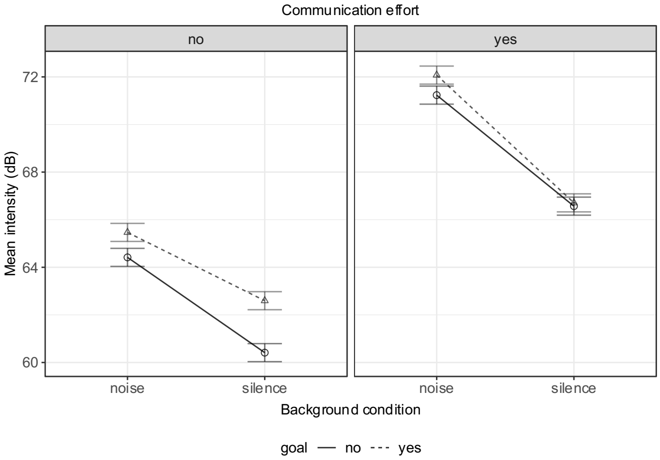


Figure 6 Reformatted `ezplot()` output for better reading.

7. Conclusions

This concludes the demonstration of how to use R and the `ez` library to carry out RM-ANOVA. RM-ANOVA is a good tool to do this kind of analysis, but it is fragile regarding sphericity. Other tools such as Linear Mixed Effects (LME) offer an alternative analysis.

Regarding the example study, it seems that communication effort increases the overall speech intensity level, in the same way that the presence of background noise did. Significant level differences were found for combinations of background masker, communication effort, and goal orientation, except when the background was quiet for communication effort tasks.

8. Acknowledgements

The author wants to thank Prof. S. Lee and Prof. J. Perkins for their comments in the preparation of the workshop and the audience at ISAPh2018 for further questions and comments.

9. References

[1] Comprehensive R Archive Network (CRAN) website [Software]. Retrieved December 7, 2018. Available from https://cran.r-project.org/.

[2] RStudio [Software]. Retrieved December 7, 2018. Available from https://www.rstudio.com/.

[3] W. Venables, D. Smith, and the R Core Team, An introduction to R. Notes on R: A programming environment for data analysis and graphics version, version 3.5.1, 2018.

[4] P. Boersma and D. Weenink, "Praat: doing phonetics by computer," 2018. Version 6.0.29. Retrieved December 7, 2018. Available from www.praat.org.

[5] M. Dowle and A. Srinivasan, data.table: Extension of data.frame, 2018. R package version 1.11.8.

[6] Y. Shue, The voice source in speech production: Data, analysis and models. PhD thesis, University of California Los Angeles, 2010.

[7] J. Villegas, I. Wilson, and J. Perkins, "Effect of task on the intensity of speech in noisy conditions," in Procedings of Acoustical Society of Japan Autumn meeting, (Aizu Wakamatsu, Japan), 2015.

[8] S. S. Mangiafico, Summary and analysis of extension program evaluation in R, version 1.13.6. New Brunswick, New Jersey: Rutgers Cooperative Extension, 2016.

[9] M. A. Lawrence, ez: Easy Analysis and Visualization of Factorial Experiments, 2015. R package version 4.3.

[10] J. Cohen, "A power primer," Psychological bulletin, vol. 112, no. 1, p. 155, 1992.

[11] H. Wickham, ggplot2: Elegant graphics for data analysis. Springer New York, 2009. http://had.co.nz/ggplot2/book.