read_me

2087153

17 April 2018

About this app

Packages Used

```
library(shiny)
library(tidyverse)
```

Calculation of the SDA statistics

Using the Equal Variance Gaussian Model.

Assumptions of the model.

Equal Variance Gaussian Model assumptions (wickens, 2001):

1. that the prob distributions of noise and signal + noise are Gaussian. This is a reasonable assumption for most types of detection experiments which should sample across large numbers of trials. Furthermore, this type of distribution is very familiar to many researchers because of its ubiquity.
2. That the mean of the noise and the noise + signal distributions are equal. Although these are rarely to be worried about as the noise and noise + signal variables can be given any centre or spread, what really matters is the relationship to each other. We cannot know the absolute values. This does not limit the predictive power of the model.
3. The variance of the two distributions are equal. This is the one most likely to be violated and may lead you to form incorrect conclusions based on a model that uses equal variance, where d' = the difference between the means, where the mean is no longer the peak of the probability distributions of noise and noise + signal distributions. There are unequal-variance models, but they are beyond my current understanding.

Hit and false alarm rate

hit rate = hits/ (hits + misses) false alarm rate = false alarms / (false alarms + correct rejections)

If either of these is equal to 1 or 0 then a slight correction is made:

- For hit rate:
  - to correct for being 0 assign it the value of 0 + 1/(2 * (hits+misses))
  - to correct for being 1 assign it the value of 1 - 1/(2 * (hits+misses))
- For false alarm rate:
  - to correct for being 0 assign it the value of 0 + 1/(2 * (false_alarms + correct_rejections))
  - to correct for being 0 assign it the value of 1 - 1/(2 * (false_alarms + correct_rejections))

False alarm and hit rates of 1 and 0 do not allow us to calculate values for d' and Beta and need to be corrected. Assigning a small value to them is an acceptable method for doing this (Lee, 2013; Wickens, 2001). Amount specified in the correction here is sourced from the TquanT SDA app (TquanT, 2017).

Beta and d’

Hit and false alarm rates are then transformed using qnorm(), giving the standard normal distribution fucntion. These are used to calculate d’:

d’ = qnorm(hit rate) - qnorm(false alarm rate)

Beta is then calculated from:

beta <- exp(-d * 0.5 * (qnorm(hit rate) + qnorm(false alarm rate)))

Criterion Beta is chosen because:

- The tool which this is based upon uses that statistic (DeBruine, n.d.).
- It is based upon the relative heights of the two probability distributions (Wickens, 2001).

Reporting requirements

Probability of hit and false alarm, as well as d’ and Beta should be reported as per the Wicken’s Elementary Signal Detection Theory book.

Both are reported in tables by the app.

Resampling

Resampled data to obtain z and p values, comparing d’ to 0 (the mean of the noise distribution, meaning that the observed results of the experiment for the signal + noise trials do not differ from those of the noise trials.)

Simulation technique based upon:

- Lessons from the research cycle course (Barr and DeBruine, 2017).
- Tuition received as part of my dissertation project work.

I struggled to get user-defined functions to work, and instead it uses a for loop for a number of iterations based upon user input. Locked at an upper limit of 10,000. Creates an empty data frame of the correct dimensions prior to the initiation of the loop.

Random samples were drawn from a choice of 0 or 1, with the same proportion as indicated by user input with the same probabilites of being drawn. This was done number of times as indicated by the user through number of iterations.

These were used to create the simulated number of hits, misses, false alarms and correct rejections needed to calculate the d’ and the Beta statistics. Using these, the standard deviations across all iterations were used, and the standardised z and p (given normal distribution assumptions) were calculated from these.

Plot Output

A standard plot of the two Gaussian probability distributions is created by the the output statistics for the model and the probability distribution functions using ggplot. This is overlaid with a vertical line indicating the Beta value, which is labelled for clarity.

There are an additional two user inputs, whose defaults are blank, which allow the user to put in an appropriate title and subtitle if they wish.

Ongoing Issues

-This is a piece of coursework for my MSc, please judge it accordingly. Updates may be slow or non-existent as a result.

-Reducing the hits and misses and/or false alarms and correct rejections causes an error that I currently don't know how to fix. I consulted the TquanT version and that also has an error (with a swanky custom error message).

-I struggled to get working user-defined functions to work, hence the repeated code and giant for loop. I know for loops can be stressful on the old computer, and can possibly go wrong and get you stuck infinitely. That's why iterations are locked at an upper limit of 10k and an empty data frame is defined prior to the initiation of the loop. Hopefully this minimises the risk.

-None of my testing has run up against these problems thus far- feel free to test it and email problems/fixes to me or fix it yourself.

- It has been observed that the plot's custom theme does not alway load in the first instance and reverts to the standard ggplot background with no border. Refreshing helps.

-It is pretty ugly and simplistic at the moment. I might add more at a later date (e.g. ROC curves, different decision criterion choices).

References

Barr D, DeBruine LM (2017) The Research Cycle. Available at: https://rgup.gitlab.io/research_cycle/index.html [Accessed March 12, 2018]

Black K. (2015) 10. Calculating p Values - R Tutorial. Available at: http://www.cyclismo.org/tutorial/R/pValues.html#calculating-a-single-p-value-from-a-normal-distribution [Accessed March 10, 2018].

Chang, W., Cheng, J., Allaire, J., Xie, Y. and McPherson, J. (2017) shiny: Web Application Framework for R. R package version 1.0.5. https://CRAN.R-project.org/package=shiny

DeBruine LM (n.d.) Face Research -Researchers -Stats -Signal Detection Analyser. Available at: http://faceresearch.org/res/stats/sda [Accessed January 8, 2018].

Dolinar S. (2014) Calculating Z-Scores using R. Available at: http://stats.seandolinar.com/calculating-z-scores-with-r/ [Accessed March 10, 2018].

Wickham, H. (2017) tidyverse: Easily Install and Load the 'Tidyverse'. R package version 1.2.1. https://CRAN.R-project.org/package=tidyverse

Lee K. (2013) SDT d prime calculation & other tips. Available at: http://www.kangleelab.com/sdt-d-prime-calculation---other-tips.html [Accessed March 22, 2018].

R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

TquanT (2017) Signal detection. Available at: https://r.tquant.eu/GrazApps/Group7_SignalDetection/ [Accessed March 22, 2018].

Wickens, T.D. (2001) Elementary Signal Detection Theory. Oxford University Press. New York.